MICROPHONE ARRAY PROCESSING TECHNIQUES FOR AUTOMATIC LECTURE MONITORING

BY

ADAM MILLER

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science in Electrical and Computer Engineering
in the College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Assistant Professor Paris Smaragdis

ABSTRACT

The gain in popularity of massive open online courses and other online educational lectures prompts the investigation of methods for automatically recording such lectures. While most previous systems in this area have utilized computer vision techniqes for tracking, we take an approach utilizing microphone arrays for both recording audio and tracking lecturers. Different source localization and source tracking methods are tested, including cross correlation and beamforming methods combined with various state space model approaches. We investigate how certain constraints granted by a lecture setting may be used to influence our tracking models, and evaluate the relative strengths and weaknesses of several possible techniques. In addition, we explore characterizations of the lecture space that allow for the microphone array to work along with a separate camera to properly record the lecturer's movement. By using the audio to track lecturers we add flexibility to the system, but also introduce difficulties in consolidating information between the microphone array and the camera. Possible methods for communication between the two are addressed, and we again find that constraints imposed by the lecture setting may be used to resolve such problems.

ACKNOWLEDGMENTS

There were many people without whom this project would not have been possible, or at the very least would have turned out much much worse. First and foremost, I'd like to thank my adviser, Paris Smaragdis, for all that he has done to help me throughout my time as an undergraduate, and for being mega-super-cool while doing it. It is because of him that I have learnt everything contained in this thesis, as well as a whole lot more (so even in the case that you, whoever you are reading this, find this thesis unsatisfactory, you should still consider this a worthwhile acknowledgment). I must say I'm not exactly sure how he was able to provide such a large proportion of what valuable knowledge I've gained as a student, but however he did it, I'm truly grateful for it.

I'd also like to thank all the graduate students at the Computational Audio Lab who have helped me throughout my time there, whether that meant working with me on a project, showing me how to do something, showing me why the thing I was doing was very wrong, or just talking to me about what cool project had been keeping them busy. This especially applies to Johannes Traa, who has spent too many of his hours helping me understand things he already knows, for which I owe him a great deal of gratitude. I'd also like to thank him for always being willing to share his encyclopedic memory of Family Guy clips and for helping me completely revamp the way I eat fruit (It's on YouTube and you should find it. It's amazing).

I've really enjoyed my time with the Lab over the past three years and I wish everyone there the best of luck in all their (certain-to-be-awesome) future endeavours. Thanks for everything guys!

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This chapter will discuss some of the motivation behind this project, as well as the goals and distinguishing features of the project. The value of automated lecture recording as a whole will be described, and an overview will be given of the techniques to be presented in the remainder of the thesis.

1.1 Overview

1.1.1 Motivation

In the last decade or so, the availability of online educational resources has increased significantly. With the launch of MIT OpenCourseWare over 10 years ago and the now growing popularity of massive open online courses (MOOCs), the internet has become a substantive educational outlet. Additionally, much of this media is comprised of lectures given in classrooms or auditoriums to real audiences, which have also been recorded and uploaded. Because of the increasing demand for such recordings, systems for efficiently recording and processing such lectures have become more and more important.

Yet the benefits of such systems can be seen even for students who attend the schools holding the lectures. In fact, many times such students are the main target for these systems, since the availability of a lecture online allows students to review that lecture at a later time [1]. Such resources free students from the burden of capturing all the material during the actual lecture and may allow them to focus more heavily on the concepts being discussed rather than writing them down for review later.

As it stands, there is a lot that goes into processing many of the open lecture resources available online. In addition to ensuring a quality recording of the lecture, there is a certain amount of editing and post-processing that usually must occur [2]. Some of post-processing is done to improve the flow of the material for the end viewer, but some may also involve dealing with mistakes made by those responsible for filming the lectures.

In addition to the time saving qualities of automated filming, such techniques can also provide substantial cost savings. While the cost of the necessary technology for these systems is continually decreasing, the cost of employing a person to record and edit the lectures is not. Because of this, prices reported for upkeep in such processes can be quite high [3], making any opportunity for automation an opportunity for savings

as well. Moreover, because automation can lower costs of producing such media, certain material may now be recorded and made available that would not have previously warranted it [4]. This increase in recorded material continues the growth of online media and further propels the positive effects of such efforts.

1.1.2 Focus

Automatic lecture monitoring as it has been described clearly can involve many separate phases. First the lecturer must take any necessary actions to initiate the lecture recording. Depending on the system, these actions may involve wearing a certain device to facilitate their tracking, setting up the proper recording equipment, or simply pressing some button on a provided interface to initiate an automated process. The specific requirements will depend on whether the system is designed to be passive or not - that is, whether or not the recorder needs to act any differently simply because the lecture is being recorded [2].

In the next step the lecture must actually be recorded – both in video and audio. Again, depending on whether the system is a passive recording system or not, doing so can range from requiring no additional action, to requiring the lecturer to take certain actions throughout the presentation to ensure proper execution of the monitoring system. Finally, as mentioned earlier, there is often post-processing that occurs in order to refine the recording.

Despite the extent of this entire process, in this thesis we will focus solely on the first aspect of this process: the recording of the speaker. More specifically, the focus will be on passive recording techniques that utilize microphone array processing for locating and tracking the speaker. A camera will then be used to film the speaker once the correct location has been determined.

In most automated lecture capture systems, computer vision methods are used on the video feeds to track the speaker [3–6]. Microphone arrays are then used to locate audience members asking questions, but are not involved in the tracking of speakers on stage [3, 5]. However, microphone array processing techniques could allow for localization of the speaker as well, and could help to add robustness to a video tracking system by providing additional information. Moreover, since many systems employing computer vision tracking methods have at least two separate cameras – one for following the speaker and one for recording the entire stage to detect movement [3, 4] – by using a microphone array to perform much, if not all, of the speaker tracking it could be possible to eliminate one camera and cut down on the costs of such systems.

CHAPTER 2

BACKGROUND

To understand the techniques used in this project, it is necessary to understand certain background material. This includes some standard microphone processing approaches as well as an introduction to state estimation. The techniques covered in this chapter will heavily influence the material in later chapters and provide a foundation for the work in the rest of the project.

2.1 Microphone Arrays

2.1.1 Properties

In this project, we will use fixed microphone arrays, which are comprised of a set of M microphones arranged in some fixed geometry. A popular example of a microphone array is the equally spaced linear array, which consists of M microphones in a line with a distance d between each consecutive microphone. An illustration of this is given in Figure 2.1, and an example of such a system in Figure 2.2a. The popularity of this configuration comes from the ease with which analysis can be performed due to the simplicity of the geometry. However, several other geometries are possible. A circular microphone array is shown in Figure 2.2b and a three dimensional "cone" geometry is shown in Figure 2.2c. In this project we will make use of the PlayStation Eye (shown in Figure 2.2a) and the Dev Audio Microcone (shown in Figure 2.2c) for our experiments.

It may not be immediately apparent how a microphone array can be useful for processing signals. At each microphone we should expect to hear the same signal with some delay, so how does this provide any advantage?

One answer to this is that the signals will not be exactly the same at each microphone; in fact, they will usually be corrupted by some noise in the environment. The redundancy of the recordings can then be used to help deduce what part of the signal is interesting, and what part of the signal is noise. This is largely the focus of beamforming techniques and will be touched on in a later section.

However, even if the signals recorded at each microphone are just shifted versions of the exact same source signal, there is still a gain to be had from using multiple microphones. This is because by using a fixed microphone array, one has infused the samples with spatial information – the information inherent in the

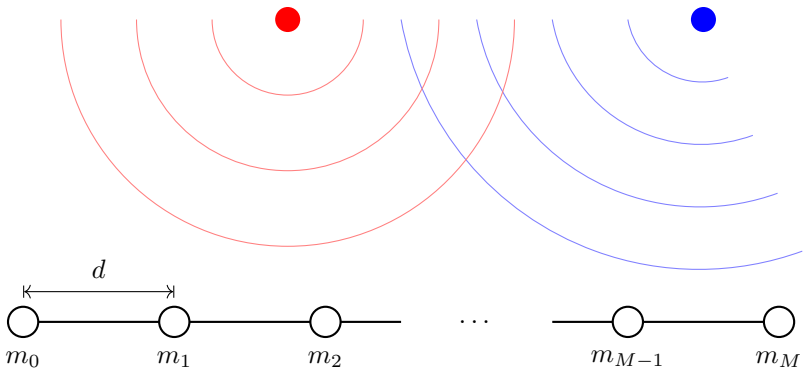
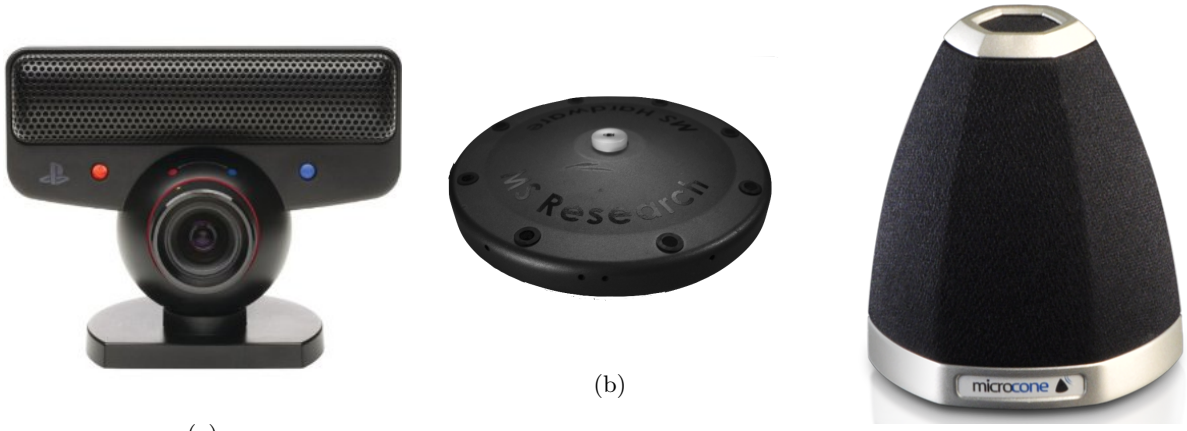


Figure 2.1: Linear microphone array configuration. Each microphone m_i lies on a line and is separated from adjacent microphones by a distance d. There are two example sound sources.



(a) (b) (c)

Figure 2.2: Microphone array examples. Figure (a) shows the PlayStation Eye, which contains a 4-microphone equally spaced linear array, (b) shows an 8-element circular array [7], and (c) shows the Dev Audio Microcone, which has six microphones on the faces of a hexagonal base and one microphone at the tip of the cone.

spatial geometry of the array. It is then possible to use this property to then infer spatial information about the source.

2.1.2 Setup

Consider a microphone array composed of M microphones with microphone i located at position $\mathbf{m}_i$. Assume that there is some source signal $s(t)$ emanating from a point source at position $\mathbf{p}$.

Now select one of the microphones to be used as reference for the others. We choose the first microphone. It is possible to calculate the amount of time it takes the sound signal to reach the reference microphone:

$$t_0 = \frac{\|\mathbf{p} - \mathbf{m}_1\|}{\nu}, \tag{2.1}$$

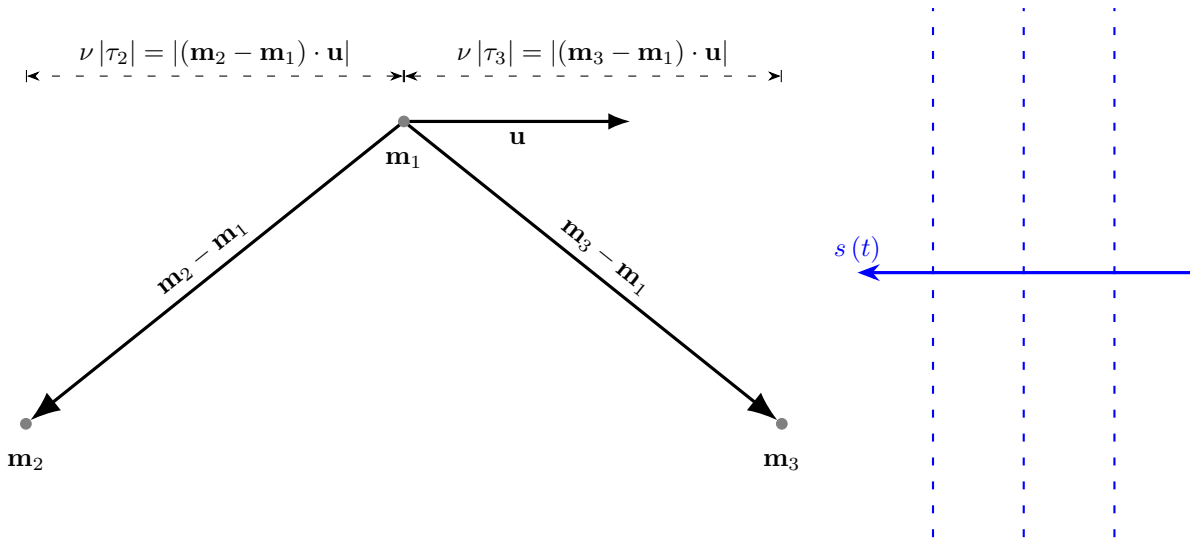


Figure 2.3: Far-field model illustration. An example microphone array consisting of three microphones. A sound source $s\,(t)$ approaches the microphone from direction of unit vector $\mathbf{u}$. Because we use the far-field model, we model the incoming sound waves as planes. The calculation of the TDOA's τ_i is illustrated.

where ν is the speed of sound. In addition to this, we can calculate the amount of time it takes for the sound to travel between the reference microphone and any other microphone. To do this we make use of the far-field model [8]. Assume the sound wave is a plane wave and $\mathbf{u}$ is a unit vector pointing in the direction of the sound source as viewed from the center of the microphone array. We then have that

$$\tau_i = \frac{(\mathbf{m}_1 - \mathbf{m}_i)\cdot\mathbf{u}}{\nu}, \tag{2.2}$$

where τ_i is the delay between the arrival of the signal at microphone 1 and the arrival of the signal at microphone i. This is known as a time difference of arrival (TDOA). This gives

$$y_i\,(t) = \alpha_i s\,(t - t_0 - \tau_i) + v_i\,(t) \tag{2.3}$$

$$= x_i\,(t) + v_i\,(t)\,, \tag{2.4}$$

where $y_i\,(t)$ is the signal recorded at microphone i, $s\,(t)$ is the unknown source signal, $x_i\,(t)$ is the attenuated source signal at microphone i, α_i are the attenuation factors, and $v_i\,(t)$ is the additive noise heard at each microphone. See Figure 2.3 for an illustration.

From this equation we see that if we assume all α_i are equal we have

$$x_i\,(t) = x_1\,(t - \tau_i)\,, \tag{2.5}$$

which reaffirms the intuitive notion that the source signal recorded at each microphone will be just a shifted version of the source signal occurring at the reference microphone.

2.2 Beamforming

2.2.1 Array Directivity

In Equation (2.5) we saw that each signal is just a shifted version of the signal at the first microphone. In turn if we want to align the signals we need only reverse this shift. To do so we transfer our focus to the frequency domain. The frequency domain equivalent of Equation (2.5) gives the following

$$X_i(f) = X_1(f)\, e^{-j2\pi f \tau_i}. \tag{2.6}$$

Now take the vector

$$\boldsymbol{\zeta}(\mathbf{v}, f) = \begin{bmatrix} e^{j2\pi f \tau_1(\mathbf{v})} & e^{j2\pi f \tau_2(\mathbf{v})} & \dots & e^{j2\pi f \tau_M(\mathbf{v})} \end{bmatrix}^T, \tag{2.7}$$

which can be seen to contain the opposite shifts for each microphone. Note that here τ_i has been written as a function of $\mathbf{v}$ to emphasize the dependence of the delays on a given steering direction. Now if we look at the vector

$$X(f) = \begin{bmatrix} X_1(f) & X_2(f) & \cdots & X_M(f) \end{bmatrix}^T, \tag{2.8}$$

we see that the vector product

$$\boldsymbol{\zeta}(\mathbf{v}, f)^T X(f) = \sum_{i=1}^{M} X_i(f)\, e^{j2\pi f \tau_i(\mathbf{v})} \tag{2.9}$$

$$= \sum_{i=1}^{M} X_i(f)\, e^{j2\pi f \frac{(\mathbf{m}_1 - \mathbf{m}_i)\cdot \mathbf{v}}{\nu}} \tag{2.10}$$

$$= \sum_{i=1}^{M} X_i(f)\, e^{-j2\pi f \frac{(\mathbf{m}_i - \mathbf{m}_1)\cdot \mathbf{v}}{\nu}} \tag{2.11}$$

$$= \sum_{i=1}^{M} X_i(f)\, e^{-j2\pi \boldsymbol{\xi}\cdot(\mathbf{m}_i - \mathbf{m}_1)} \tag{2.12}$$

equates to a spatial Fourier transform, where

$$\boldsymbol{\xi} = \frac{f\mathbf{v}}{\nu} \tag{2.13}$$

is the spatial frequency vector. Just as a spectral Fourier transform can be interpreted as determining the energies in each constituent frequency of a signal, this spatial Fourier transform can be viewed as determining the energies in each constituent direction of the signal, given a specific frequency. This result makes sense as here we sample the signal across space where as in the spectral version we do so across time.

Note however that the above equations were written assuming a fixed source direction, $\mathbf{u}$. In reality, the

directivity is also a function of the source direction as we have

$$X_i(\mathbf{u}, f) = X_1(f)\, e^{-j2\pi f \tau_i(\mathbf{u})}, \tag{2.14}$$

giving us the complete directivity

$$\begin{aligned}
\mathcal{D}(\mathbf{v}, \mathbf{u}, f) &= \boldsymbol{\zeta}(\mathbf{v}, f)^T X(\mathbf{u}, f) && (2.15)\\
&= \sum_{i=1}^{M} X_i(\mathbf{u}, f)\, e^{j2\pi f \tau_i(\mathbf{v})} && (2.16)\\
&= X_1(f) \sum_{i=1}^{M} e^{-j2\pi f \tau_i(\mathbf{u})} e^{j2\pi f \tau_i(\mathbf{v})} && (2.17)\\
&\propto \sum_{i=1}^{M} e^{-j2\pi f \tau_i(\mathbf{u})} e^{j2\pi f \tau_i(\mathbf{v})}. && (2.18)
\end{aligned}$$

What we have shown here is that by aligning the signals as if they came from a certain direction and then combining them, we are able to attribute an amount of the signal that came from that direction. That is, we perform a spatial Fourier transform. We now see an alternative explanation for the same process and how it falls into the greater context of beamforming.

2.2.2 Delay and Sum Beamformer

In the previous section we found how to the determine the directivity of an array as a function of a steering direction $\mathbf{v}$, source direction $\mathbf{u}$, and frequency f. Consider now the case where $\mathbf{v} = \mathbf{u}$. If we steer the array in this direction, we see that we get

$$\begin{aligned}
\boldsymbol{\zeta}(\mathbf{u}, f)^T X(\mathbf{u}, f) &= \sum_{i=1}^{M} X_i(\mathbf{u}, f)\, e^{j2\pi f \tau_i(\mathbf{u})} && (2.19)\\
&= X_1(f) \sum_{i=1}^{M} e^{-j2\pi f \tau_i(\mathbf{u})} e^{j2\pi f \tau_i(\mathbf{u})} && (2.20)\\
&= M X_1(f), && (2.21)
\end{aligned}$$

which is just the signal at the reference microphone multiplied by the number of microphones in the array.

We can of course view this as simply summing all the recorded signals after shifting them the correct amount to offset the propagation delay. If we do this with the raw recorded signals and average the result

we get

$$z(t) = \frac{1}{M}\sum_{i=1}^{M} y_i(t+\tau_i) \tag{2.22}$$

$$= \frac{1}{M}\sum_{i=1}^{M} \alpha_i x_i(t+\tau_i) + \frac{1}{M}\sum_{i=1}^{M} v_i(t+\tau_i) \tag{2.23}$$

$$= \alpha s(t-t_0) + \frac{1}{M}\sum_{i=1}^{M} v_i(t+\tau_i), \tag{2.24}$$

where

$$\alpha = \frac{1}{M}\sum_{i=1}^{M} \alpha_i. \tag{2.25}$$

Quite appropriately, this is known as the delay and sum method for beamforming [7–9]. In cases where the $v_i(t)$ are uncorrelated, it can increase the signal-to-noise ratio (SNR) by a factor of M, as a result of averaging the noise across channels. However, in the case where the noise at the microphones are correlated, the noise reduction decreases and with perfect correlation is nonexistent [9].

The delay and sum method is the most straightforward of a variety of beamforming approaches, all of which aim to use information available in the signal recordings to reduce unwanted noise and spatially steer the array [7–9].

2.3 State Space Models

2.3.1 Definition

State space models (SSM) provide a means for estimating the unknown state of a system $\mathbf{x}_t$ at time t given a sequence of observations $\mathbf{z}_{1:t}$ [10, 11]. To do so, we define the dynamics of the system as follows:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t), \tag{2.26}$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t). \tag{2.27}$$

Here the function $f(\cdot)$ describes how the states evolve, given the previous state $\mathbf{x}_{t-1}$ and some noise $\mathbf{u}_t$ whose statistics are known. The function $h(\cdot)$ describes the emissions of the observations given the current current state $\mathbf{x}_t$ and some observation noise $\mathbf{v}_t$ whose statistics are also known. To completely describe this system we must also define some initial distribution for $\mathbf{x}_1$:

$$\mathbf{x}_1 \sim p(\mathbf{x}_1|\boldsymbol{\pi}), \tag{2.28}$$

where $\boldsymbol{\pi}$ is a set of parameters that determines the initial distribution. We can now define the transition

equation, Equation (2.26), to dictate how we expect our state to evolve and the emission equation, Equation (2.27), to dictate how we expect our observations to occur.

It is important to note that because our initial state is a random variable and the states and observations are continually contaminated by random noise, we actually can describe the transition and emission equations by distributions:

$$\mathbf{x}_t \sim p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right), \tag{2.29}$$

$$\mathbf{z}_t \sim p\left(\mathbf{z}_t|\mathbf{x}_t\right), \tag{2.30}$$

which instead gives us transition and emission distributions. The two notations are equivalent, as the transition and emission distributions in Equation (2.29) and Equation (2.30) are fully described by the transition and emission equations in Equation (2.26) and Equation (2.27). However, thinking about the evolution of the system in this manner will prove useful as we investigate ways of forming our estimations in a Bayesian context.

2.3.2 Recursive Bayesian Filtering

As shown in the previous section, our system can be fully described by a set of recursive equations. Moreover, the transitions and observations are described by a set of probability distributions. We now approach the problem in the context of recursive Bayesian filtering [10–12].

Our goal is to maintain some belief in our state $\mathbf{x}_t$ given all the values we've observed, $\mathbf{z}_{1:t}$. This involves maintaining the posterior distribution $p\left(\mathbf{x}_t|\mathbf{z}_{1:t}\right)$. We can use the recursive nature of the system to do this recursively at each time t, where our process occurs in two distinct steps.

First, in the *predict* step, we look at the prediction of the next state given all the previous observations:

$$p\left(\mathbf{x}_t|\mathbf{z}_{1:t-1}\right) = \int p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) p\left(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}\right)\, \mathrm{d}\mathbf{x}_{t-1}. \tag{2.31}$$

We see that to do this we need our previous posterior $p\left(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}\right)$, and the transition distribution described in Equation (2.29). Next, in the *update* step, we update our prediction with the knowledge of the current observation to get our new posterior at time t:

$$p\left(\mathbf{x}_t|\mathbf{z}_{1:t}\right) = p\left(\mathbf{x}_t|\mathbf{z}_t, \mathbf{z}_{1:t-1}\right) \tag{2.32}$$

$$= \frac{p\left(\mathbf{z}_t|\mathbf{x}_t\right) p\left(\mathbf{x}_t|\mathbf{z}_{1:t-1}\right)}{p\left(\mathbf{z}_t|\mathbf{z}_{1:t-1}\right)} \tag{2.33}$$

$$\propto p\left(\mathbf{z}_t|\mathbf{x}_t\right) p\left(\mathbf{x}_t|\mathbf{z}_{1:t-1}\right). \tag{2.34}$$

Here we use our prediction $p\left(\mathbf{x}_t|\mathbf{z}_{1:t-1}\right)$ from Equation (2.31) and the emission distribution described by

Equation (2.30). This allows us to think of the two steps as predicting the next state and then updating our prediction by weighting it with the likelihood of our observation[1].

Unless certain assumptions hold, the integral given in Equation (2.31) cannot be computed analytically. Luckily, various methods for dealing with this problem exist and will be addressed as we use them later in the text.

[1]While we refer to this as the "predict, update" process, other variations do exist, such as "predict, correct" or "imagine, punish". However, the underlying process is the same for all.

CHAPTER 3

DIRECTION OF ARRIVAL ESTIMATION AND STATELESS TRACKING

This chapter will explore the direction of arrival estimation methods as they were implemented in our system. It will also describe how these estimates were used to perform stateless tracking of speakers. By stateless, we refer to the lack of a state dynamics model for the system or the maintenance of any such external state that would allow for a prediction of the next state.

3.1 Direction of Arrival Estimation

We first consider methods for estimating the direction of arrival (DOA) of a sound. As described in Section 2.1.2, this equates to estimating the TDOA values given by Equation (2.2).

3.1.1 GCC-PHAT

The first method we describe is the generalized cross correlation (GCC) method [9, 13, 14]. We seek to find

$$\hat{\tau}_{i,j} = \underset{\tau}{\operatorname{argmax}} \, \Psi_{i,j}(\tau), \tag{3.1}$$

where

$$\Psi_{i,j}(\tau) = \int_{-\infty}^{\infty} \vartheta(f) \, Y_i^*(f) \, Y_j(f) \, e^{j2\pi f \tau} \, \mathrm{d}f \tag{3.2}$$

$$\tag{3.3}$$

is the generalized cross correlation function between microphone i and microphone j. Here $\vartheta(f)$ is a spectrum weighting function that can be used to scale the signal spectra and improve the accuracy of the resulting TDOA estimate. This can be interpreted as an additional set of filters through which the signals pass that can help shape them for better estimation [13, 14]. We see that for a unity weighting function the GCC is

equivalent to the standard cross correlation function

$$R_{i,j}(\tau) = \int_{-\infty}^{\infty} Y_i^*(f)\, Y_j(f)\, e^{j2\pi f\tau}\, \mathrm{d}f \tag{3.4}$$

$$= \int_{-\infty}^{\infty} y_i(t-\tau)\, y_j(t)\, \mathrm{d}t. \tag{3.5}$$

A variety of weighting functions are available [9, 13, 14], however one of the most popular is the phase transform (PHAT) weighting

$$\vartheta_{\mathrm{PHAT}}(f) = \frac{1}{|Y_i^*(f)\, Y_j(f)|}, \tag{3.6}$$

which has been shown to be quite robust in noisy and reverberant environments [15]. The combination of the generalized cross correlation method with the phase transform weighting is referred to as GCC-PHAT.

3.1.2 Point Estimates

Using the GCC-PHAT method, we can obtain a set of $\hat{\tau}_{i,j}$ values by searching for a peak in the resulting GCC function of each necessary microphone pair. This set comprises our TDOA estimates. From this, using Equation (2.2) we can set up the following system:

$$\frac{1}{\nu}\begin{bmatrix} \mathbf{m}_1 - \mathbf{m}_2 \\ \mathbf{m}_1 - \mathbf{m}_3 \\ \vdots \\ \mathbf{m}_1 - \mathbf{m}_M \end{bmatrix} \hat{\mathbf{u}} = \begin{bmatrix} \hat{\tau}_2 \\ \hat{\tau}_3 \\ \vdots \\ \hat{\tau}_M \end{bmatrix}, \tag{3.7}$$

with

$$\hat{\tau}_i = \hat{\tau}_{1,i} \tag{3.8}$$

to be consistent with notation in Equation (2.2). Note that here we only consider delays between microphone 1 and all others since other delays become linearly dependent. The solution may be solvable if the number of microphones is one more than the number of dimensions in the search space and the microphones have an appropriate geometry, which will make the matrix on the left of Equation (3.7) invertible. Otherwise we must use a least squares solution.

However, this method has a few negatives. First off, by doing this we will be getting point estimates for the direction of arrival with no measure of likelihood. Therefore, we are making a hard decision about our estimate and if this estimate is used in a larger system, we have made a great commitment at an early stage. This should usually be avoided if at all possible. Secondly, to actually implement this method we must discretize the τ search space. In doing so, we are inherently discretizing the DOA search space of $\hat{\mathbf{u}}$,

which may end up leading to a discretization that is inefficient or ineffective for our purposes. While efficient discretizations and search methods exist for localization [16], we will not pursue them and will instead favor a simpler solution.

A remedy for the first problem could be to aggregate the correlation values for each combination of discrete τ values. Then the magnitude of these correlation values could be used as a likelihood score as we will discuss later. However, this would be extremely expensive computationally, and in doing so we would be considering many infeasible directions. Yet, despite this problem, this method is not far off. In fact if we perform a similar calculation for only combinations of τ_i values that correspond to feasible search directions, we can rectify both of the mentioned problems.

3.1.3 Likelihood Approach

As mentioned, the goal is to obtain a likelihood for each feasible direction of arrival. We use methods similar to [17, 18]. Unfortunately the space of feasible direction of arrivals is continuous and while attempts to create continuous likelihood models over the search space based on TDOA estimates exist [19], we choose to instead discretize the search space for simplicity. By sampling our search space, we can now determine exactly which TDOAs are of interest. That is, for each feasible source direction $\mathbf{v}$ we can calculate $\tau_i(\mathbf{v})$ for any microphone i. We then need only compute the correlation values:

$$\Psi_{i,j}\left(\tau_j(\mathbf{v}) - \tau_i(\mathbf{v})\right) = \int_{-\infty}^{\infty} \vartheta(f)\, Y_i^*(f)\, Y_j(f)\, e^{j2\pi f(\tau_j(\mathbf{v})-\tau_i(\mathbf{v}))}\, \mathrm{d}f. \tag{3.9}$$

Now to get a likelihood score we sum across the different channel pairs, giving

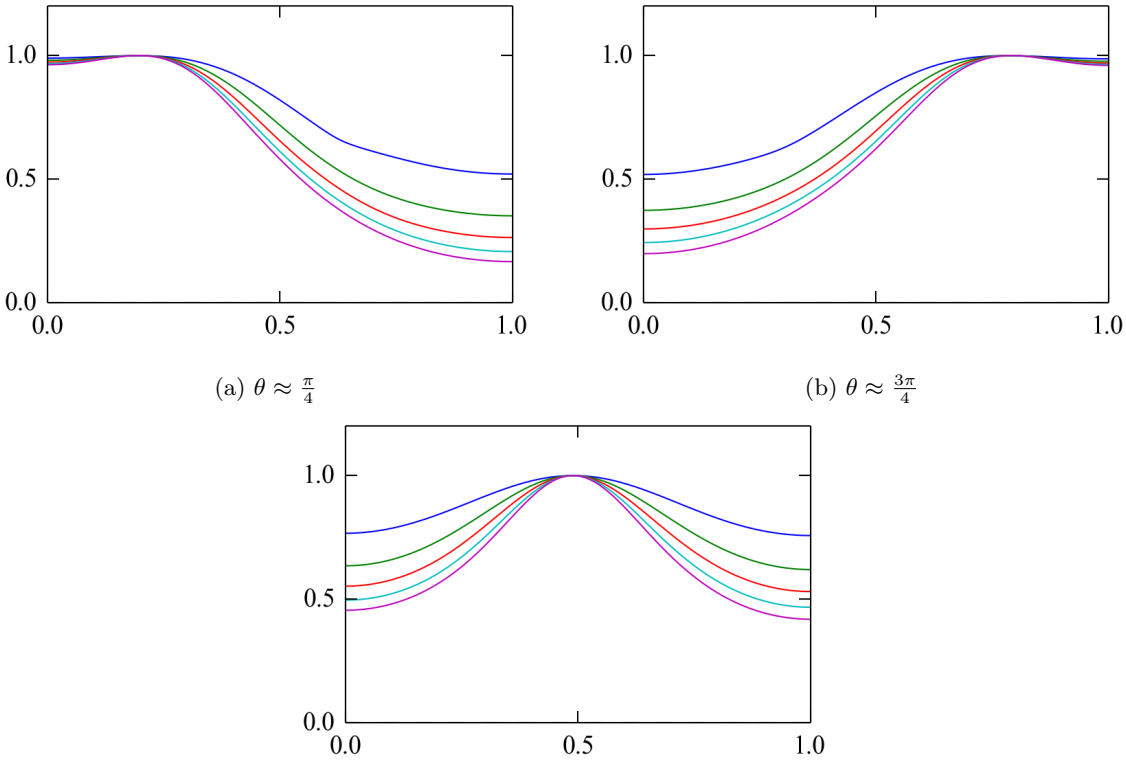
$$\mathcal{L}_{\mathrm{GCC}}(\mathbf{v}) = \sum_{i=1}^{M} \sum_{j=i+1}^{M} \Phi\left(\Psi_{i,j}\left(\tau_j(\mathbf{v}) - \tau_i(\mathbf{v})\right)\right), \tag{3.10}$$

where $\Phi(x) = x^k$ is a likelihood shaping function, used to sharpen peaks and reduce sidelobes [20]. See Figure 3.1 for examples of this likelihood distribution for the PlayStation Eye linear array.

3.1.4 Beamforming and SRP-PHAT

The method described is very closely related to another method known as SRP-PHAT (where SRP stands for steered response power) when the PHAT weighting function is used to calculate the GCC [17]. This results from the fact that we are calculating the energy in the scaled crosspower spectrum [15] after steering the array in each feasible direction $\mathbf{v}$. This suggests that this method can be viewed in a beamforming context.

Consider using a delay and sum beamformer to steer the signal in each feasible direction $\mathbf{v}$ and then taking



(a) $\theta \approx \frac{\pi}{4}$

(b) $\theta \approx \frac{3\pi}{4}$

(c) $\theta \approx \frac{\pi}{2}$

Figure 3.1: GCC likelihoods. Normalized GCC Values vs. normalized source DOA azimuthal angle $\frac{\theta}{\pi}$. Various values of k in the shaping function $\Phi(x) = x^k$ are plotted. We see the peak sharpen as the value of k increases from 1 to 5. Data was recorded by a PlayStation Eye with a single source one meter from the array.

the power of the corresponding signal:

$$\mathcal{L}_{\text{DS}}(\mathbf{v}) = \int_{-\infty}^{\infty} \left[\sum_{i=1}^{M} y_i(t + \tau_i(\mathbf{v}))\right]^2 dt \tag{3.11}$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{M} \int_{-\infty}^{\infty} y_i(t + \tau_i(\mathbf{v}))\, y_j(t + \tau_j(\mathbf{v}))\, \mathrm{d}t \tag{3.12}$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{M} \int_{-\infty}^{\infty} y_i(t + \tau_i(\mathbf{v}) - \tau_j(\mathbf{v}))\, y_j(t)\, \mathrm{d}t \tag{3.13}$$

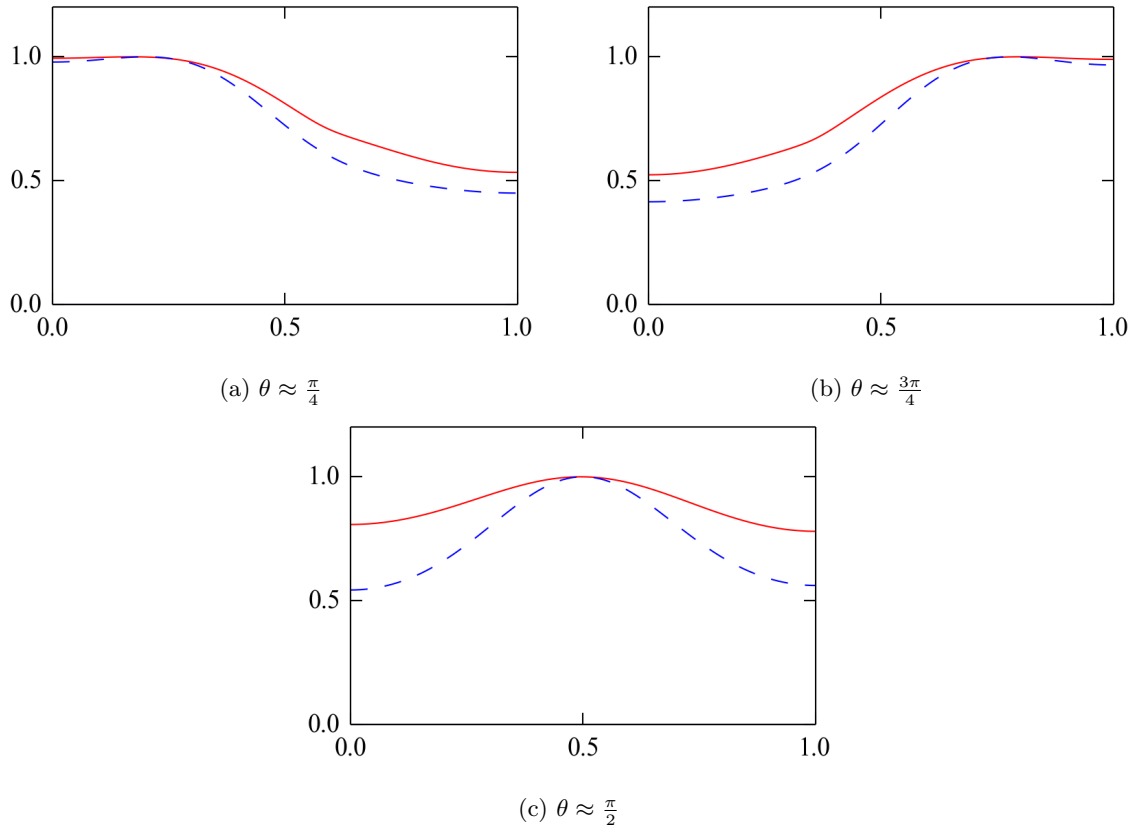
$$= \sum_{i=1}^{M}\sum_{j=1}^{M} \int_{-\infty}^{\infty} Y_i^*(f)\, Y_j(f)\, e^{j2\pi f(\tau_j(\mathbf{v}) - \tau_i(\mathbf{v}))}\, \mathrm{d}f \tag{3.14}$$

$$= 2\mathcal{L}_{\text{GCC}}(\mathbf{v}) + \sum_{i=1}^{M} \int_{-\infty}^{\infty} Y_i^*(f)\, Y_i(f)\, \mathrm{d}f. \tag{3.15}$$

We see that if we use $\Phi(x) = x$ and add in a weighting function $\vartheta(f)$ we get that the result is equal to

twice our earlier likelihood plus the total signal energy. It can be shown that these two likelihood scores are closely related to a Bayesian likelihood of the signal given the prospective direction [21].

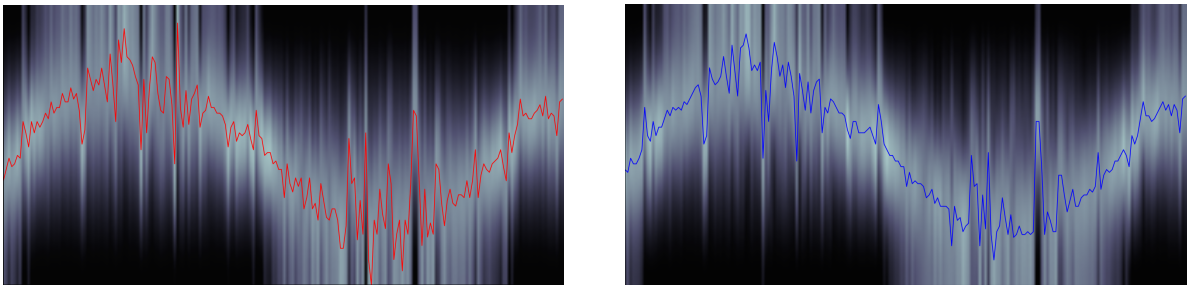
Just as a spectrum weighting function $\vartheta(f)$ was added with the GCC method, a weighting function for combining frequencies can be added here. While there again are various weighting functions, the phase transform is often used, leading to the denomination of this method as SRP-PHAT [18]. A comparison of the SRP-PHAT results and GCC-PHAT results is given in Figure 3.2. There also exist analogous SRP algorithms for beamforming methods other than delay and sum. Good coverage of these methods is given in [7].



(a) $\theta \approx \frac{\pi}{4}$ (b) $\theta \approx \frac{3\pi}{4}$

(c) $\theta \approx \frac{\pi}{2}$

Figure 3.2: GCC-PHAT vs. SRP-PHAT functions. The solid line shows the normalized likelihood obtained using GCC-PHAT with an identity shape function, while the dashed line shows that obtained using SRP-PHAT. We see the two are quite similar, though the SRP-PHAT appears to be more peaked. However, by using a different coefficient in the shaping function with GCC-PHAT the peaks could be easily accentuated. The data was recorded on the PlayStation Eye with the speaker one meter from the array.



(a) (b)

Figure 3.3: GCC-PHAT and SRP-PHAT source tracking. DOA Angle vs. time for the GCC-PHAT likelihood method (a) and SRP-PHAT likelihood method (b). The normalized likelihoods are represented by the shading, with lighter regions corresponding to higher likelihoods. The estimate across each time frame is plotted, which can be seen to correspond to peaks in the likelihood function. Data was recorded with a PlayStation Eye and a single source speaking one meter away while varying the angle to the array.

3.2 Tracking

3.2.1 Raw DOA Estimates

With DOA likelihoods available, we can now track sources by using the likelihoods to form an estimate at each time frame. While this would have been possible with only point estimates as well, the likelihoods will prove useful later when our models become more complex.

However, a quick glance at Figure 3.3 will display the major problem with simply using unprocessed DOA estimates to track a source. As is evident from the plots, both methods provide quite noisy estimates, with SRP-PHAT doing slightly better in this regard.

Luckily, this problem is not an unfamiliar one in the signal processing domain and is simply one of optimal filtering, as will be discussed next.

CHAPTER 4

STATE SPACE TECHNIQUES FOR CONSTRAINED SOURCE TRACKING

This chapter will now consider source tracking methods based on state space models where the speaker is constrained in their movement. This constraint will allow us to adapt certain models that would otherwise be difficult to apply. Unlike the previous chapter, these methods will make use of a model of the sound source's movement in order to improve location estimates. For an overview of state space models, see Section 2.3.

4.1 Speaker Constraints

4.1.1 Motivation

We will find that for certain techniques, the ability to constrain the speaker's movement in some way will prove very helpful in developing our models. In order to do this, we consider the natural constraints on a speaker during a lecture. In the vast majority of situations, the speaker tends to stay on a plane in the front of the room that extends vertically out of the ground. Whether it is on a stage, or at the front of a classroom, their movement is usually limited from side to side at the front of the room and they tend not to walk out into the audience. By using this constraint, we can equate DOA estimates to positions on the plane in three-dimensional space, which will provide us with more freedom in employing established techniques.

4.1.2 Setup

To do this, we define this plane to be all $\mathbf{s}$ such that

$$0 = \mathbf{n}^T(\mathbf{s} - \mathbf{o}), \tag{4.1}$$

where $\mathbf{n}$ is the normal vector for the plane (parallel to the floor), and $\mathbf{o}$ is some point on the plane.

Now for a DOA $\mathbf{v}$ from the microphone array, we have that the direction in the coordinates of the room will be $\mathbf{Uv}$, where $\mathbf{U}$ is a matrix whose columns form the basis of the microphone array coordinate system. If the microphone array is located at $\mathbf{r}$ we have that the point on the plane in that direction must satisfy

$$0 = \mathbf{n}^T(\mathbf{r} + t\mathbf{Uv} - \mathbf{o}) \tag{4.2}$$

for some t. We can solve for t, which gives the corresponding point $\mathbf{s}(\mathbf{v})$ to be

$$\mathbf{s}(\mathbf{v}) = \mathbf{r} + \frac{\mathbf{n}^T(\mathbf{o}-\mathbf{r})}{\mathbf{n}^T\mathbf{U}\mathbf{v}}\mathbf{U}\mathbf{v}. \tag{4.3}$$

4.2 Grid-based Methods

The most straightforward approach to applying recursive Bayesian filtering (see Section 2.3.2) to the DOA filtering problem is to use grid-based methods, as outlined in [11]. These methods involve discretizing the state space into a grid of distinct states $\mathbf{x}_t^{(i)}$ at each time t, where $i = 1, 2, \ldots, N_s$, with N_s as the number of discrete states in the state space. If we have that $w_{t|t}^{(i)} = \Pr\left(\mathbf{x}_t = \mathbf{x}_t^{(i)} | \mathbf{z}_{1:t}\right)$, then we can estimate the posterior pdf at time t as a sum of weighted point estimates:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \sum_{i=1}^{N_s} w_{t|t}^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right). \tag{4.4}$$

By substituting Equation (4.4) into Equation (2.31) and Equation (2.33) we get the new predict and update equations:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \sum_{i=1}^{N_s} w_{t|t-1}^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right), \tag{4.5}$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \sum_{i=1}^{N_s} w_{t|t}^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right), \tag{4.6}$$

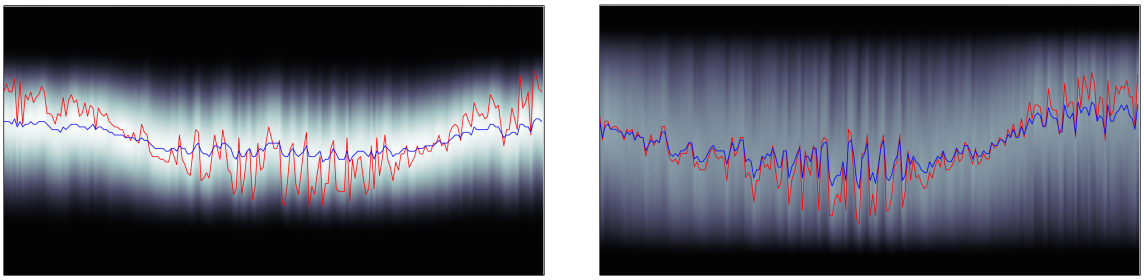
where

$$w_{t|t-1}^{(i)} = \sum_{j=1}^{N_s} w_{t-1|t-1}^{(j)} p\left(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(j)}\right), \tag{4.7}$$

$$w_{t|t}^{(i)} = \frac{w_{t|t-1}^{(i)} p\left(\mathbf{z}_t | \mathbf{x}_t^{(i)}\right)}{\sum_{j=1}^{N_s} w_{t|t-1}^{(j)} p\left(\mathbf{z}_t | \mathbf{x}_t^{(j)}\right)}. \tag{4.8}$$

Since we were already discretizing our DOA space, we should be able to use this method quite easily. In our case, each $\mathbf{x}_t^{(i)}$ is a unit vector pointing in a prospective DOA. We can choose our emission distribution $p(\mathbf{z}_t|\mathbf{x}_t)$ to be either of the likelihood functions we defined in Section 3.1.3 and Section 3.1.4. However, we must still setup the dynamics of our state transitions. That is, we must define $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. But because of the constraints we've imposed, if we define our dynamics on the plane $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = p_{\text{plane}}(\mathbf{s}_t|\mathbf{s}_{t-1})$ we have our transition distribution to be

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = p_{\text{plane}}\left(\mathbf{s}(\mathbf{x}_t)\,|\mathbf{s}(\mathbf{x}_{t-1})\right). \tag{4.9}$$



(a) $\sigma_x^2 = .5$ (b) $\sigma_x^2 = 4$

Figure 4.1: Grid tracking. The estimate posterior distribution is plotted at each time step with lighter regions corresponding to more probable directions. The blue plot is the DOA estimate using the grid method, while the red line is the unfiltered estimate. Both cases use a Gaussian transition distribution around the current state as the emission distribution. As can be seen, the higher state variance in (b) than (a) allows the estimate to move more freely but also causes more noise in the estimate. Recorded with a PlayStation Eye and speaker staying on a plane 2 meters from the array.

This allows us to assess how a speaker will move within the room instead of how they will appear to move to the microphone array, which should help to develop a more accurate model. Using this, we can now perform Bayesian filtering using the grid based method. An example of results is given in Figure 4.1.

There unfortunately is a pretty major problem with this method. Namely, the filter performs very poorly when the sound source is located off center from the point of view of the array. As can be seen in Figure 4.1, once the estimates are off the center the error in the grid based estimate gets larger. This has to do with the sparse sampling of the $\mathbf{s}(\mathbf{x}_t)$ space as $|\mathbf{x}_t \cdot \mathbf{n}|$ gets smaller. At a certain point, $\mathbf{s}\left(\mathbf{x}_t^{(i+1)}\right)$ is very far away from $\mathbf{s}\left(\mathbf{x}_t^{(i)}\right)$ for some i. Because of this, $w_{t|t-1}^{(i+1)}$ becomes so much smaller than $w_{t|t-1}^{(i)}$ that even though it may be that $p\left(\mathbf{z}_t|\mathbf{x}_t^{(i+1)}\right) > p\left(\mathbf{z}_t|\mathbf{x}_t^{(i)}\right)$, it still ends up that $w_{t|t}^{(i)} > w_{t|t}^{(i+1)}$. Therefore, at a certain point the posterior stops being able to follow the source as they move further and further from the array along the plane.

One way to help the posterior more accurately track the state is by allowing a greater variance in movement of the source along the plane. This helps to keep $w_{t|t-1}^{(i+1)}$ and $w_{t|t-1}^{(i)}$ closer together, which allows the posterior to achieve peaks further off center. This can be seen in comparing the two cases in Figure 4.1. Because Figure 4.1b uses a higher variance in movement, the estimate follows the DOA more closely as it veers to the side. The downside of this, of course, is that with the higher assumed variance in movement we must compromise our system model and the filtered estimates are far more noisy. Another resolution could be to attempt to sample the plane uniformly instead sampling the DOA state space uniformly. However, we will not investigate this approach and will instead discuss other favorable techniques.

Despite the problems, in cases where the change in speaker location is not very large, this method may be sufficient. As can be seen in Figure 4.1a, the filtered estimates are less noisy than the raw DOA estimates. And if we use a stationary grid, much of the calculation can be carried out ahead of time so that at each time step only a matrix vector product and element wise vector multiplication are necessary to get the updated

posterior.

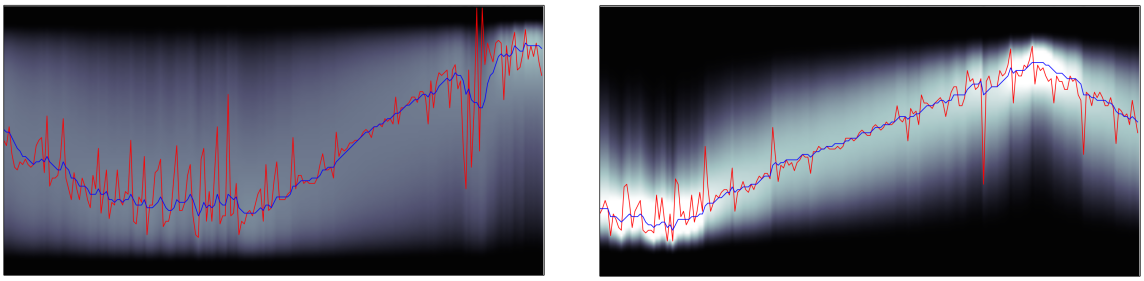
4.3 Kalman Filtering

If we assume that $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{z}_t|\mathbf{x}_t)$ are both Gaussian and $f(\cdot)$ and $h(\cdot)$ are both linear in their arguments, then Equation (2.31) and Equation (2.33) have closed form solutions and we get the famous Kalman filter [10, 22–24]. The Kalman filter is a very well established filtering algorithm and is optimal given the previous assumptions. However, in order to use it in this context we will have to again utilize the constraints on our system. This is necessary because the state transition distribution of a DOA state will certainly be non-Gaussian, seeing as it is a function of an angular variable, and is thus circular in some sense.

Additionally, even if our state were a location that varied linearly in the surrounding space, the transformation into the DOA search space would be non-linear and thus would violate the Kalman filtering assumptions. While there do exist techniques for performing Kalman filtering on the DOA's or TDOA's using extended Kalman filter techniques [23, 25] or wrapped Kalman filter techniques [26], we will not consider them here.

Once again, we consider the speaker to be constrained to a plane. However, where before we were able to use the SRP likelihoods from Section 3.1.3 and Section 3.1.4 directly as the emission probabilities, we now use these likelihoods to find the maximum likelihood estimate of the DOA, $\hat{\mathbf{u}}$, and then we use $\mathbf{s}(\hat{\mathbf{u}})$ as the observation $\mathbf{z}_t$ in our Kalman filtering algorithm. In this way, both the states $\mathbf{x}_t$ and the observations $\mathbf{z}_t$ can be described in terms of linear transformations and Gaussian distributions upon on the plane.

We see in Figure 4.2 that the Kalman filter performs much better than the grid-based method. Because we did not need to discretize our state space, the Kalman filter has no problem following the source as it gets further and further from the array. We can also see that the filtering greatly reduces the noise associated with the raw DOA estimates. As justification for transforming into a linear space before applying the filter, in Figure 4.3 we show the posterior distribution and DOA likelihood at a single time frame for two different source directions and distances to the plane. Note that the posterior distribution is quite non-Gaussian, especially when the DOA is off center. In turn it would have been incorrect to use the standard Kalman filter directly on the DOA's.



(a) (b)

Figure 4.2: Kalman filter tracking. In both (a) and (b), the blue plot represents the Kalman filter estimate, while the red plot represents the unfiltered estimate. The DOA posterior is plotted vertically in each frame with lighter regions being more probable. In (a) the plane is 2 meters from the array and in (b) the plane is 4 meters from the array.

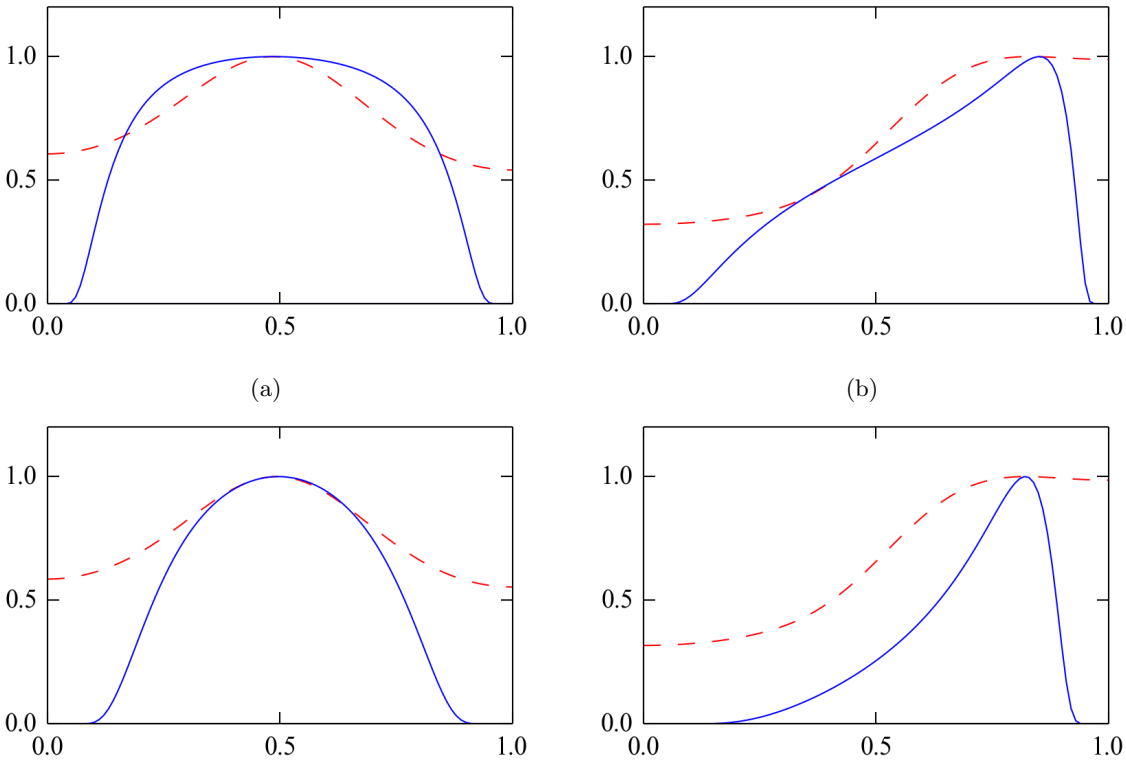


Figure 4.3: Kalman filter projected distributions. All plots show the normalized DOA posterior distribution (solid) which results from projecting the Gaussian posterior state distribution from the plane onto the unit circle. This gives clearly non-Gaussian results. The normalized SRP likelihood (dashed) is also plotted. The x axis shows the normalized DOA azimuthal angle. In (a) and (b) we have the plane 2 meters from the array and in (c) and (d) 4 meters from the array. (a) and (c) correspond to a DOA azimuthal angle of about $\frac{\pi}{2}$ and (b) and (d) correspond to a DOA azimuthal angle of about $\frac{3\pi}{4}$

CHAPTER 5

STATE SPACE TECHNIQUES FOR UNCONSTRAINED SOURCE TRACKING

In the previous chapter we explored methods for tracking a speaker granted we could constrain the speaker's movement to lie on a predefined plane in three-dimensional space. Here we explore methods that forgo this assumption and instead focus on tracking the speaker in DOA space. To do so, we must constrain our DOA's to lie on the unit sphere. That is, for any prospective direction $\mathbf{v}$ we have that $\|\mathbf{v}\| = 1$. We must enforce this as we have no reliable information about the distance of the source from the microphone array, which comes from our using the far-field model. This will introduce nonlinearities that prevent the use of Kalman filtering and thus will require more versatile methods.

5.1 Particle Filtering

5.1.1 Overview

As we saw in Section 4.2, it is possible to approximate the posterior pdf $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ as a sum of weighted point estimates:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) \approx \sum_{l=1}^{L} w_t^{(l)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(l)}\right), \tag{5.1}$$

where L is the number of point estimates, or particles. However, as we discussed, there were many drawbacks to the grid based method that fell out of the fact that we were using a predefined discretization of the state space. In this way, our point estimates were not moving, but simply had their weights being constantly updated using the newest available observations.

An alternative class of methods allows the locations of the particles in the state space to vary, which eliminates many of the problems associated with the grid-based approaches. These methods are known as Sequential Monte Carlo Methods, or Particle Filters, and continually generate new particles given the locations of the old ones [10, 11, 27, 28].

5.1.2 Predict

Just like the previous methods discussed, the particle filter fits into the recursive Bayesian filtering framework. The main difference lies in how the predict step is carried out. In the predict step, we generate

a new prediction of the posterior given in Equation (5.1) by sampling from some proposal distribution $q\left(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}\right)$ over our state space to generate new particle locations. Thus we have for $l = 1, \ldots, L$:

$$\tilde{\mathbf{x}}_t^{(l)} \sim q\left(\mathbf{x}_t|\mathbf{x}_{1:t-1}^{(l)}, \mathbf{z}_{1:t}\right). \tag{5.2}$$

The goal is to use a distribution that makes use of as much information as we have available to us (given computational constraints) to generate samples in the state space that will be close to the actual state value. As discussed in [27], the optimal sampling distribution is $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(l)}, \mathbf{z}_t\right)$. Intuitively this makes sense, as it takes into consideration all available information up to time t and remains faithful to the transition model of the system. However, because of the practical difficulties often associated with using this distribution, the transition distribution $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(l)}\right)$ is frequently used instead. How the sampling is actually carried out depends on the specific distribution, but extensive research in Monte Carlo methods exists for dealing with such problems [24, 29, 30].

5.1.3 Update

In the update portion of the filter we must now correctly re-weight the particles so that our posterior estimate remains as accurate as possible. The weighting process follows from a technique known as importance sampling [10, 24]. We update the weights using

$$w_t^{(l)} = w_{t-1}^{(l)} \frac{p\left(\mathbf{z}_t|\tilde{\mathbf{x}}_t^{(l)}\right) p\left(\tilde{\mathbf{x}}_t^{(l)}|\mathbf{x}_{t-1}^{(l)}\right)}{q\left(\tilde{\mathbf{x}}_t^{(l)}|\mathbf{x}_{1:t-1}^{(l)}, \mathbf{z}_{1:t}\right)} \tag{5.3}$$

and then re-normalize so that

$$\sum_{l=1}^{L} w_t^{(l)} = 1. \tag{5.4}$$

Note that if the transition distribution is used as the proposal distribution, i.e.

$$q\left(\tilde{\mathbf{x}}_t^{(l)}|\mathbf{x}_{1:t-1}^{(l)}, \mathbf{z}_{1:t}\right) = p\left(\tilde{\mathbf{x}}_t^{(l)}|\mathbf{x}_{t-1}^{(l)}\right), \tag{5.5}$$

then we simply have that

$$w_t^{(l)} \propto w_{t-1}^{(l)} p\left(\mathbf{z}_t|\tilde{\mathbf{x}}_t^{(l)}\right). \tag{5.6}$$

5.1.4 Resample

One of the problems with the algorithm as presented is that after a while, the particles will have dispersed throughout the state space so that for most values of l, $w_t^{(l)} \approx 0$. Thus, the posterior estimate will be dominated by a few select particles. To overcome this, the particles are resampled from the estimated

posterior distribution and their weights renormalized:

$$\mathbf{x}_t^{(l)} \sim \sum_{i=1}^{L} w_t^{(i)} \delta\left(\mathbf{x}_t - \tilde{\mathbf{x}}_t^{(i)}\right), \tag{5.7}$$

$$w_t^{(l)} = \frac{1}{L}. \tag{5.8}$$

This resampling step may be carried out at every time step t or only once the number of appreciable particles has fallen below some threshold. The algorithm outlined here is known as the Sequential Importance Resampling (SIR) algorithm [27].

5.1.5 Estimates

Form the nature of the particle filter, we see that using the particles as a direct estimate of a probability density function would be somewhat inaccurate – the density would consist only of several weighted spikes, with nothing in between. While we could place weighted kernels at the particle locations, this is not really necessary for our purposes. In fact, we are not actually interested in evaluating the posterior probability at arbitrary locations, but rather forming estimates of the distribution's statistics.

The most important estimate will be the expected state given by the posterior:

$$\mathbb{E}[\mathbf{x}_t|\mathbf{z}_{1:t}] \approx \sum_{l=1}^{L} p\left(\mathbf{x}_t|\mathbf{z}_{1:t}\right) \mathbf{x}_t^{(l)} \approx \sum_{l=1}^{L} w_t^{(l)} \mathbf{x}_t^{(l)}. \tag{5.9}$$

This formula is used widely in importance sampling techniques and when valid can be shown to be an unbiased estimate of the true expectation. In addition, higher order moments may be estimated in a similar manner. However, this may not be valid for some distributions, as it is possible the result no longer lies in the state space. This may occur when the state space is the surface of a sphere, as will be used shortly. Nevertheless, in our case straightforward modifications are possible.

5.2 von Mises-Fisher Particle Filter (vMFPF)

In order to track sources on the unit sphere we make use of the von Mises-Fisher distribution when in three dimensions [31] and the von Mises distribution [24] when in two dimensions. As both distributions result from conditioning a Gaussian distribution on the unit sphere in the corresponding number of dimensions, we will refer mostly to the von Mises-Fisher distribution with the knowledge that the techniques can be applied in any number of dimensions. The vMFPF is presented in [32], and much of the work to follow is based on methods presented there.

5.2.1 Model Definition

For our state transition model, We define the transition and emission distributions to be

$$\mathbf{x}_t \sim v\mathcal{MF}\left(\mathbf{x}_{t-1}, \kappa_u\right), \tag{5.10}$$

$$\mathbf{z}_t \sim v\mathcal{MF}\left(\mathbf{x}_t, \kappa_v\right), \tag{5.11}$$

where $v\mathcal{MF}\left(\mu, \kappa\right)$ is a von Mises-Fisher distribution with mean μ and concentration κ, κ_u is the concentration parameter of the state distribution, and κ_v is the concentration parameter of the emission distribution. From these equations, we see that we are taking the observed direction to be the actual direction contaminated by von Mises-Fisher distributed noise. The observed direction can be deduced from the recorded signals using one of the DOA estimation methods described in Chapter 3.

To form our state estimates, we make a straightforward modification of the formula given in Equation (5.9). As described in [31], the maximum likelihood estimate of μ for a set of samples $\mathbf{x}_i$ drawn from a von Mises-Fisher Distribution is given by

$$\hat{\mu} = \frac{\sum_i \mathbf{x}_i}{\|\sum_i \mathbf{x}_i\|}, \tag{5.12}$$

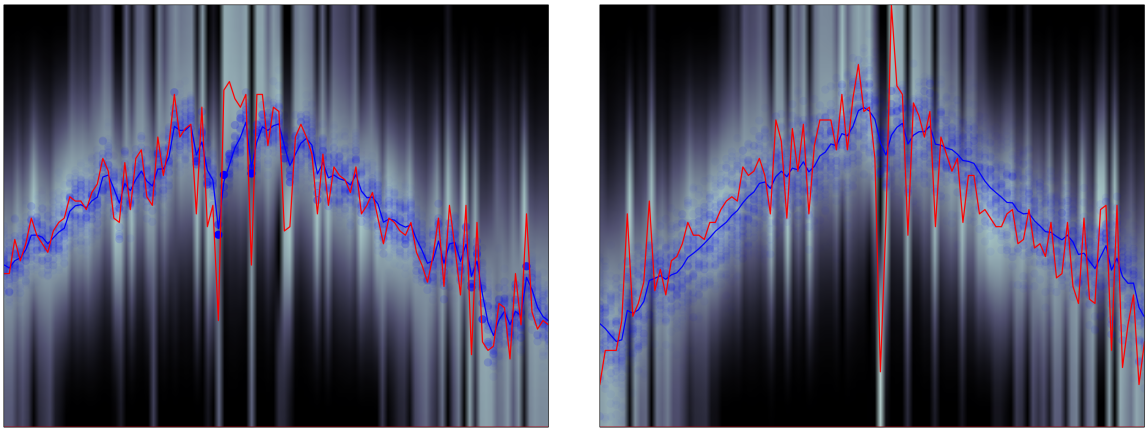
which can be seen to be a projection of the standard result onto the unit sphere. Therefore, to compute our estimate we use

$$\mathbb{E}[\mathbf{x}_t|\mathbf{z}_{1:t}] \approx \frac{\sum_{l=1}^{L} p\left(\mathbf{x}_t^{(l)}|\mathbf{z}_{1:t}\right)\mathbf{x}_t^{(l)}}{\left\|\sum_{l=1}^{L} p\left(\mathbf{x}_t^{(l)}|\mathbf{z}_{1:t}\right)\mathbf{x}_t^{(l)}\right\|} \approx \frac{\sum_{l=1}^{L} w_t^{(l)}\mathbf{x}_t^{(l)}}{\left\|\sum_{l=1}^{L} w_t^{(l)}\mathbf{x}_t^{(l)}\right\|}. \tag{5.13}$$

5.2.2 Performance

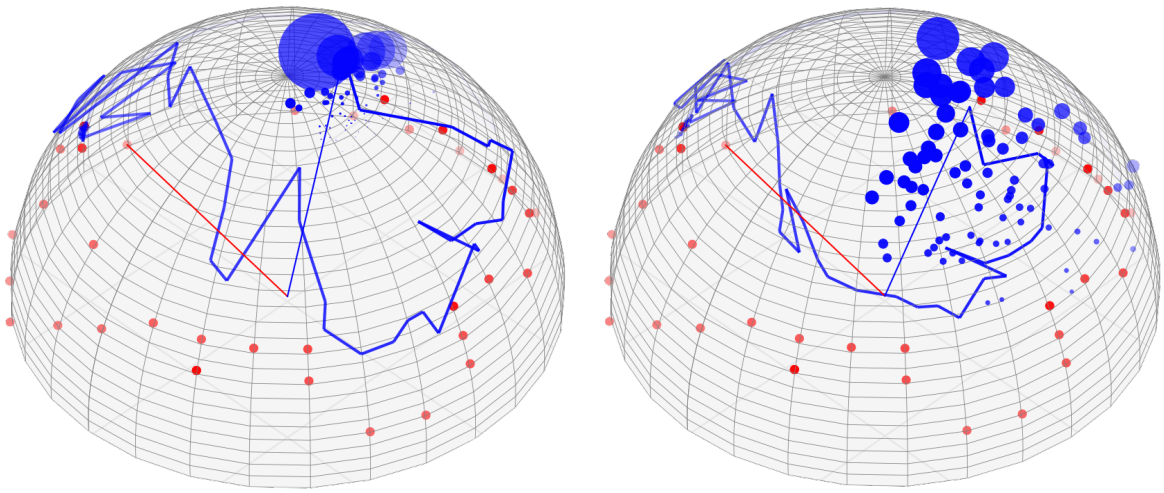
Examples of the performance of this model are given in Figure 5.1 for the two-dimensional case and Figure 5.2 for the three-dimensional case. In each figure, two different values of the observation concentration parameter κ_v are shown. It is clear that the two-dimensional performs much better. However, we see that in both the two-dimensional and three-dimensional case, the higher concentration in Figure 5.1a and Figure 5.2a leads to a narrower estimate of the posterior. Consequently, the estimated states will follow the observations more closely as is confirmed in Figure 5.1a. In this way, a lower concentration can be used to smooth the state estimate as can be seen in Figure 5.1b.

One of the problems with spreading the noise distribution to smooth estimates is that state estimates begin to lag behind the observations. This can be seen in Figure 5.1b. One way to mitigate this could be to include velocity components in the state space by either tracking the rotation of the state vector [32] or through the use of quarternions as described in [33]. However, since the model of lecturer movement we are concerned with will likely not contain substantial velocity, we will ignore the velocity consideration as it is unlikely to provide advantages worth the extra complexity.



(a) $L = 50, \kappa_u = 100, \kappa_v = 25$ (b) $L = 50, \kappa_u = 100, \kappa_v = 5$

Figure 5.1: von Mises particle filter tracking. The red line shows the DOA point observations, while the blue line shows the filtered state estimates. The blue dots represent the particles with each particles weight being indicated by its transparency. The darker a particle, the larger the weight. Results were recorded on a Playstation Eye microphone array with a speaker roughly one meter from the array.



(a) $L = 80, \kappa_u = 100, \kappa_v = 25$ (b) $L = 80, \kappa_u = 100, \kappa_v = 5$

Figure 5.2: von Mises-Fisher particle filter tracking. The red dots show the DOA point observations, with the red vector based at the origin of the hemisphere pointing to the current observation. The blue line across the surface of the hemisphere shows the filtered state estimates, with the blue line based at the origin of the hemisphere pointing to the current estimate. The blue dots represent the particle locations at the current time frame, with their size representing the particle weights. The larger a particle is, the more heavily weighted it is. Results were recorded using a Dev Audio a meter from the array.

5.2.3 Problems

Perhaps the biggest drawback of this model is its inability to effectively cope with spurious DOA estimates. While this is not as evident when tracking on the unit circle, tracking on the unit sphere enlarges the space of DOA observations and allows for far more noise to show up in the system. This is shown in Figure 5.2, where we see the sporadic nature of the observation points as well as the inaccuracy of the state estimates as they fall quite far from many of the pertinent observations.

A close look at Figure 5.2 will also reveal the cause of this problem. The current DOA observation (at the tip of the red vector emanating from the center of the plot) is on the other half of the hemisphere than most of the particles. Because of this we see that the particles closer to the observation have much larger weights and in turn draw the estimate heavily toward what was a spurious observation.

Much of the problem results from how we have chosen our emission distribution. That is, we have a mismatch between our model and our data. Because we have assumed that our observations are distributed in unimodal fashion about our state, any outlier observation will heavily sway our belief in the underlying state of our system. Clearly this emission distribution is not accurate. While we get an abundance of DOA observations near our state, we also get a large amount of outliers that show up. To deal with this, we revise our model.

5.3 von Mises-Fisher Switching Particle Filter (vMFSPF)

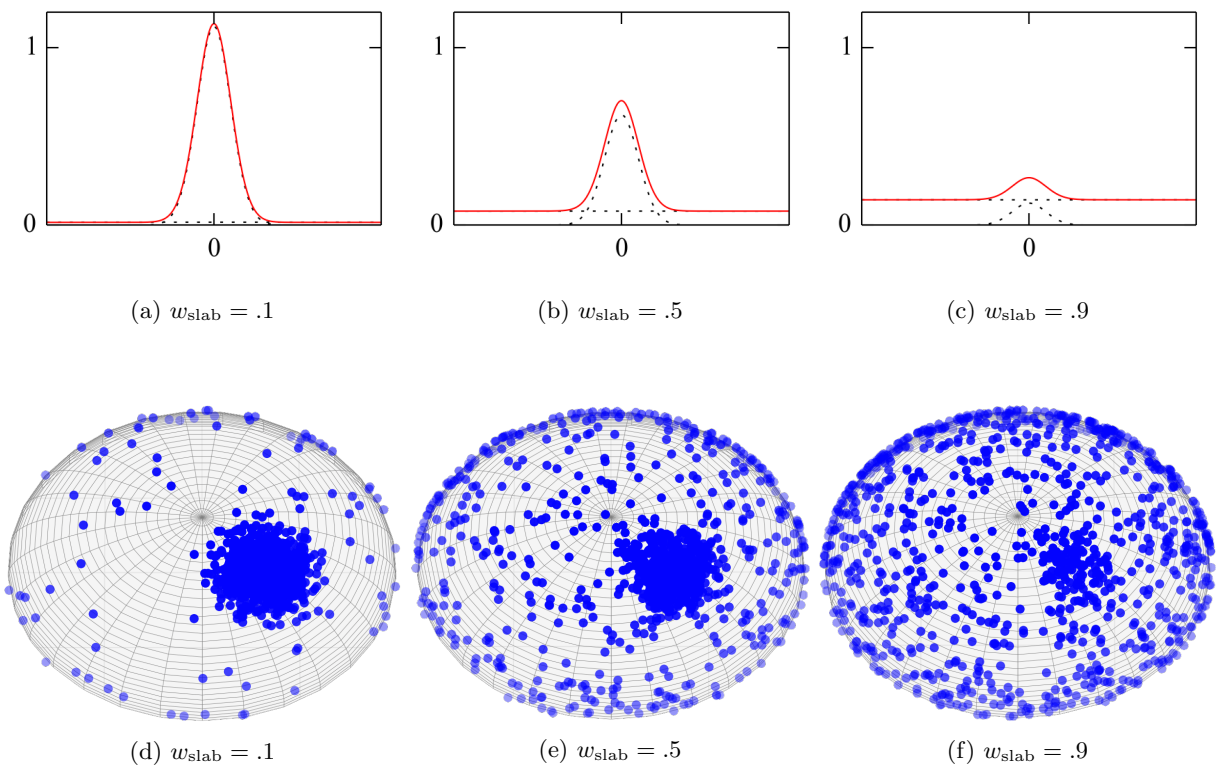
5.3.1 Spike and Slab

Because of the downfalls with the unimodal emission density described, we must instead employ a model more faithful to our data. To do this, we adopt what has been referred to as a "spike and slab" distribution [34] for our emission density. This distribution will consist of the weighted sum of a unimodal "spike" distribution and a flat "slab" distribution, both centered over our current estimate. Because we are tracking in bounded spaces, we can actually use a uniform distribution for the slab. See Figure 5.3 for illustrations of such distributions.

Such a distribution over our state should more accurately represent our emissions. While DOA observations should hover around the current state $\mathbf{x}_t$, there will also be random outliers. Now we can attribute such outliers to the background "slab" distribution and leave our state estimate as is.

5.3.2 Switching State Space Models

To make use of this mixture distribution, we will modify our system model. In addition to a state $\mathbf{x}_t$ and observation $\mathbf{z}_t$ at each time t we also add a discrete switching variable c_t. This gives us the following overall



(a) $w_{\text{slab}} = .1$ (b) $w_{\text{slab}} = .5$ (c) $w_{\text{slab}} = .9$

(d) $w_{\text{slab}} = .1$ (e) $w_{\text{slab}} = .5$ (f) $w_{\text{slab}} = .9$

Figure 5.3: Spike and slab distributions. The "spike and slab" distributions are plotted for three different values of the slab weight. In (a), (c), and (b), spikes are von Mises distributions with $\mu = 0$, $\kappa = 10$. The dotted lines represent the individual components while the solid line represents the overall mixture distribution. In (d), (e), and (f), spikes are von Mises-Fisher distributions with $\kappa = 80$. In all cases, slabs are uniform distributions over the entire support.

model:

$$\mathbf{x}_t \sim p\left(\mathbf{x}_t | \mathbf{x}_{t-1}\right), \tag{5.14}$$

$$c_t \sim \Pr\left(c_t | c_{t-1}\right), \tag{5.15}$$

$$\mathbf{z}_t \sim p\left(\mathbf{z}_t | \mathbf{x}_t, c_t\right). \tag{5.16}$$

This is one example of a switching state space model [10, 35]. The switching variable can be viewed as a means of switching between different observation models. For $c_t = k$, we have that $\mathbf{z}_t \sim p\left(\mathbf{z}_t | \mathbf{x}_t, c_t = k\right)$, which can be defined to be a different distribution for each value of $k = 1, \ldots, K$. This is commonly used to model sensor failure where c_t represents whether a sensor is defective or not. This allows for dealing with erroneous observations by establishing a different emission distribution to use [36]. While in our case we will only have two values for c_t (which indicate either the spike or slab distributions to be active), we see that the general model can account for any number of possible values. However the complexity of the computations will grow as K^2, making very large support over c_t costly.

If we view our model in a generative sense, we can think of our state estimates $\mathbf{x}_t$ as evolving just as before. However, at each time t, a value of c_t is generated that then determines how the observation is emitted from the given state $\mathbf{x}_t$.

5.3.3 Switching Particle Filter

To deal with the switching variable in our model, we must modify our particle filtering algorithm as in [37]. Where before we sought to estimate the posterior pdf $p(\mathbf{x}_t|\mathbf{z}_{1:t})$, we now seek to estimate the joint posterior $p(\mathbf{x}_t, c_t|\mathbf{z}_{1:t})$. We have that

$$p(\mathbf{x}_t, c_t|\mathbf{z}_{1:t}) = \sum_{k=1}^{K} p(\mathbf{x}_t, c_t = k|\mathbf{z}_{1:t})\, \delta(c_t - k). \tag{5.17}$$

Thus we need to estimate $p(\mathbf{x}_t, c_t = k|\mathbf{z}_{1:t})$ for each value of $k = 1, \ldots, K$. This gives

$$p(\mathbf{x}_t, c_t = k|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t, c_t = k, \mathbf{z}_{1:t-1})\, p(\mathbf{x}_t, c_t = k|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{5.18}$$

$$= \frac{p(\mathbf{z}_t|\mathbf{x}_t, c_t = k) \Pr(c_t = k|\mathbf{x}_t, \mathbf{z}_{1:t-1})\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{5.19}$$

$$= \frac{p(\mathbf{z}_t|\mathbf{x}_t, c_t = k) \Pr(c_t = k|\mathbf{z}_{1:t-1})\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}. \tag{5.20}$$

We see that this equation fits nicely into the Bayesian filtering framework. Where before we had one predictive density $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$, we now have two: $\Pr(c_t = k|\mathbf{z}_{1:t-1})$ and $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$.

The first of the two we may compute analytically:

$$\Pr(c_t = k|\mathbf{z}_{1:t-1}) = \sum_{j=1}^{K} \Pr(c_t = k|c_{t-1} = j) \Pr(c_{t-1} = j|\mathbf{z}_{1:t-1}) \tag{5.21}$$

$$= \sum_{j=1}^{K} \pi_{j,k} \Pr(c_{t-1} = j|\mathbf{z}_{1:t-1}). \tag{5.22}$$

We only need the transition probability $\pi_{j,k} = \Pr(c_t = k|c_{t-1} = j)$ defined by our model and our previous estimate $\Pr(c_{t-1} = k|\mathbf{z}_{1:t-1})$. If we define

$$r_{k,t} = \Pr(c_t = k|\mathbf{z}_{1:t}) \tag{5.23}$$

and

$$r^{-}_{k,t} = \Pr(c_t = k|\mathbf{z}_{1:t-1}), \tag{5.24}$$

we can write this prediction relationship as

$$r_{k,t}^{-} = \sum_{j=1}^{K} \pi_{j,k} r_{j,t-1}. \tag{5.25}$$

Our second distribution, $p\left(\mathbf{x}_t|\mathbf{z}_{1:t-1}\right)$, is the standard predictive distribution from Equation (2.31) and will require sequential importance sampling as before. Thus we modify the weighting updates given in Equation (5.3) to fit Equation (5.20), giving

$$w^{*(l)}_{k,t} = w^{(l)}_{t-1} \frac{p\left(\mathbf{z}_t|\tilde{\mathbf{x}}^{(l)}_t, c_t = k\right) p\left(\tilde{\mathbf{x}}^{(l)}_t|\mathbf{x}^{(l)}_{t-1}\right) r^{-}_{k,t}}{q\left(\tilde{\mathbf{x}}^{(l)}_t|\mathbf{x}^{(l)}_{1:t-1}, \mathbf{z}_{1:t}\right)}, \tag{5.26}$$

where $w^{*(l)}_{k,t}$ are the unnormalized particle weights for $c_t = k$. We then normalize the weights to get

$$w^{(l)}_{k,t} = \frac{w^{*(l)}_{k,t}}{\sum_{m=1}^{K}\sum_{n=1}^{L} w^{*(n)}_{m,t}}. \tag{5.27}$$

We can now complete the updates with

$$w^{(l)}_{t} = \sum_{k=1}^{K} w^{(l)}_{k,t} \tag{5.28}$$

and

$$r_{k,t} = \sum_{l=1}^{L} w^{(l)}_{k,t}\ . \tag{5.29}$$

For the resampling stage we still sample using Equation (5.7), but in addition to normalizing all $w^{(l)}_t$ as in Equation (5.8), we also set $r_{k,t} = \frac{1}{K}$ for all k.

5.3.4 Model Definition

Now that we have a framework for a switching particle filter, we define the model for use in our DOA tracking. Out state transition distribution stays the same as that given in Equation (5.10).

For our observations model we must define $p\left(\mathbf{z}_t|\mathbf{x}_t, c_t\right)$. We allow our switching variable c_t to take on two values: $c_t = 1$ indicates the observation will come from the spike in our emission distribution, while $c_t = 2$ indicates the observation will come from the slab. Thus we get

$$\mathbf{z}_t \sim p\left(\mathbf{z}_t|\mathbf{x}_t, c_t\right) = \sum_{k=1}^{K} p\left(\mathbf{z}_t|\mathbf{x}_t, c_t = k\right) \delta\left(c_t - k\right) \tag{5.30}$$

with

$$p\left(\mathbf{z}_t|\mathbf{x}_t, c_t = k\right) = \begin{cases} v\mathcal{MF}\left(\mathbf{z}_t; \mathbf{x}_t, \kappa_v\right) & k = 1, \\ U_{\text{sphere}}\left(\mathbf{z}_t\right) & k = 2, \end{cases} \tag{5.31}$$

where κ_v is the observation concentration parameter as before and U_{sphere} is the uniform distribution over the unit sphere.

For our switching variable transitions, we define a transition model π that gives all transition probabilities $\pi_{j,k} = \Pr\left(c_t = k|c_{t-1} = j\right)$. This gives us

$$c_t \sim p\left(c_t|c_{t-1}; \pi\right). \tag{5.32}$$

Our π parameter lets us define the respective weights for the spike and slab in our distribution over time. In the simplest case, we may set

$$\pi_{j,k} = \begin{cases} \Pr\left(c_t = k\right) & k = j, \\ 0 & \text{otherwise}, \end{cases} \tag{5.33}$$

which breaks any dependence of c_t on c_{t-1} and lets us define constant priors for our switching variables. In this case, $\Pr\left(c_t = 1\right)$ corresponds to the spike weight and $\Pr\left(c_t = 2\right)$ to the slab weight. The effects of different weightings on the emission distribution were shown before in Figure 5.3. These priors allow us to tune our observation model to match our data.

5.3.5 Estimate Behavior

As usual, we intend to use our particle filter to estimate the true state $\mathbf{x}_t$ at time t. Because our model has already taken into consideration the switching observation models, we need not do anything different and can simply use the formula given in Equation (5.13).

However, expanding the formula to make the dependence on c_t explicit will help in understanding how the filter works. For the general case with K switching values, after omitting the normalization imposed

when calculating von Mises-Fisher expectations we get

$$\mathbb{E}[\mathbf{x}_t|\mathbf{z}_{1:t}] \approx \sum_{l=1}^{L} w_t^{(l)} \mathbf{x}_t^{(l)} \tag{5.34}$$

$$= \sum_{l=1}^{L} \sum_{k=1}^{K} w_{k,t}^{(l)} \mathbf{x}_t^{(l)} \tag{5.35}$$

$$\approx \sum_{k=1}^{K} \sum_{l=1}^{L} p\left(\mathbf{x}_t, c_t = k|\mathbf{z}_{1:t}\right) \mathbf{x}_t^{(l)} \tag{5.36}$$

$$= \sum_{k=1}^{K} \sum_{l=1}^{L} p\left(\mathbf{x}_t|c_t = k, \mathbf{z}_{1:t}\right) \Pr\left(c_t = k|\mathbf{z}_{1:t}\right) \mathbf{x}_t^{(l)} \tag{5.37}$$

$$\approx \sum_{k=1}^{K} \mathbb{E}[\mathbf{x}_t|c_t = k, \mathbf{z}_{1:t}] \Pr\left(c_t = k|\mathbf{z}_{1:t}\right). \tag{5.38}$$

From this we see that by using the given estimator our estimate is comprised of a weighted sum of the estimates conditioned on each of the switching values. This is the behavior we hope for, as it is the behavior of the exact expectation. It also makes intuitive sense, as we are essentially finding the estimate for each observation model separately, and then weighting each with the likelihood of that model given the observations. Furthermore, from above we see that

$$\mathbb{E}[\mathbf{x}_t|c_t = k, \mathbf{z}_{1:t}] \Pr\left(c_t = k|\mathbf{z}_{1:t}\right) = \sum_{l=1}^{L} w_{k,t}^{(l)} \mathbf{x}_t^{(l)} \ , \tag{5.39}$$

which shows how the weighted expectations show up in our updates.

Now consider this within the context of the spike and slab model. To see how this filtering method deals with noise and spurious observations, we consider two cases. The first occurs when the DOA observation $\mathbf{z}_t$ is near the particles $\mathbf{x}_t^{(l)}$, and the second when the DOA observation $\mathbf{z}_t$ is far from the particles $\mathbf{x}_t^{(l)}$. For a visual depiction, see Figure 5.4.

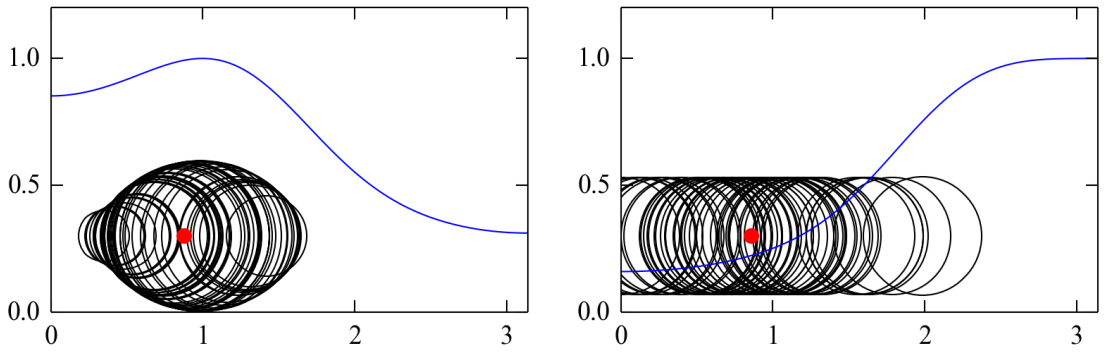
In the first case, depicted in Figure 5.4a, we will have that

$$p\left(\mathbf{z}_t|\mathbf{x}_t, c_t = 1\right) \gg p\left(\mathbf{z}_t|\mathbf{x}_t, c_t = 2\right). \tag{5.40}$$

Therefore, if we use the simple transition model given in Equation (5.33) and $\frac{\Pr(c_t=2)}{\Pr(c_t=1)}$ is not too large, we will have

$$w_t^{(l)} = w_{1,t}^{(l)} + w_{2,t}^{(l)} \tag{5.41}$$

$$\approx w_{1,t}^{(l)}, \tag{5.42}$$



(a) (b)

Figure 5.4: von Mises switching particle filter weights. Particle filter before and after a quick change in location by the speaker. Particles $\mathbf{x}_t^{(l)}$ and their weights $w_t^{(l)}$ are shown, along with the resulting state estimate. The black circles are centered at the particle locations, with the size of the circle growing with the weight of the corresponding particle. The red dot shows the location of the estimate. The SRP likelihood is plotted, with the DOA observation falling at the peak of the likelihood function. Recorded on a Playstation Eye.

which means that

$$\mathbb{E}[\mathbf{x}_t|\mathbf{z}_{1:t}] \approx \mathbb{E}[\mathbf{x}_t|c_t = 1, \mathbf{z}_{1:t}]\Pr(c_t = 1). \tag{5.43}$$

This is confirmed in Figure 5.4a as we see that the state estimate is very near the DOA observation. If we assume that the observation came from the spike in our emission density, this is the correct estimate. Additionally, we see that the weights conform to a unimodal shape about the state estimate, as we expect from Equation (5.42).

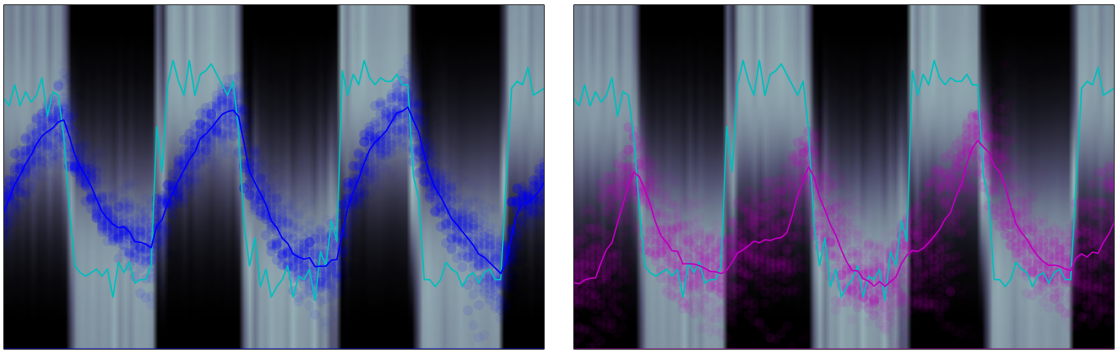
For the second case, depicted in Figure 5.4b, by the same analysis we have that

$$\mathbb{E}[\mathbf{x}_t|\mathbf{z}_{1:t}] \approx \mathbb{E}[\mathbf{x}_t|c_t = 2, \mathbf{z}_{1:t}]\Pr(c_t = 2). \tag{5.44}$$

Because $p(\mathbf{z}_t|\mathbf{x}_t, c_t = 2)$ is a uniform distirbution, we get that $w_t^{(l)} \approx w_{2,t}^{(l)}$ are uniform and the state estimate will lie at the center of the particle group. The estimate should undergo little movement as long as the weights remain uniform, as the particles will continue to spread about randomly and symmetrically (assuming the proposal density form Equation (5.2) is symmetric, which is a reasonable assumption given our state transition distribution is symmetric). Thus as long as the observation is far from the particles, the state estimate will remain relatively static. This can be seen in Figure 5.4b, where the DOA observation location has changed greatly from Figure 5.4a, while the state estimate has not.

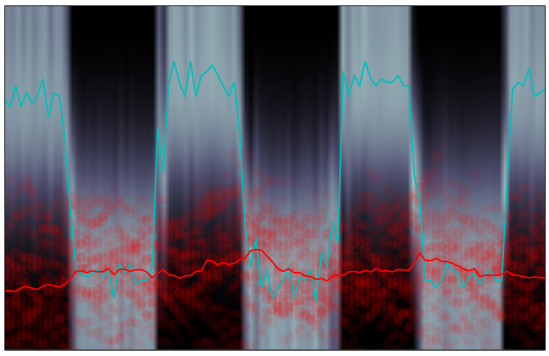
This is the behavior we hope for, and makes perfect sense. If an observation is much more likely to have resulted from our background distribution, we don't want it to influence our state estimate. Yet the Monte Carlo nature of the algorithm ensures that if the observation is not spurious, eventually enough particles will

come close enough to the observation that the state estimate will update to the new location. For a display of this behavior over time, see Figure 5.5. Note that when the observation is far from the particle group, the variance of the particles and thus the variance of our estimate will greatly increase. However, this increase in variance follows from our assumption that the most probable emission density in that case provides no spatial clue to the location of the underlying state, and is thus expected.



(a) $\Pr(c_t = 1) = 1$

(b) $\Pr(c_t = 1) = .5$



(c) $\Pr(c_t = 1) = .1$

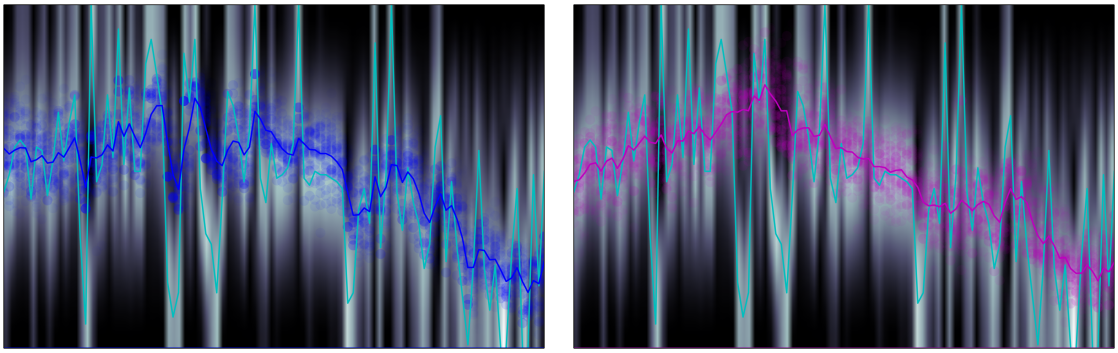
Figure 5.5: Effect of switching variable priors in von Mises switching particle filter. Particles and estimates are given for three different prior probabilities on the "spike" observation model. All three plots use the same recorded data from a Playstation Eye. The DOA observations are given by the cyan plot. Again, heavier weighted particles are more opaque. While this pattern of movement is not addressed by our state transition model, this display is still useful for seeing how quickly changing observations affect our estimates and how the priors over c_t factor in. Note that the case of $\Pr(c_t = 1) = 1$ in (a) reduces to the standard particle filter. Here $\kappa_u = 100$ and $\kappa_v = 5$.

5.3.6 Performance

We see the tracking performance of the switching particle filter for the two-dimensional and three-dimensional cases in Figure 5.6 and Figure 5.7 respectively. While the two-dimensional case with switching does show improvement over the standard particle filter algorithm (given in Figure 5.6a) the improvement in more pronounced in the three-dimensional case. Again, this results from the larger space and thus greater variation of observations.

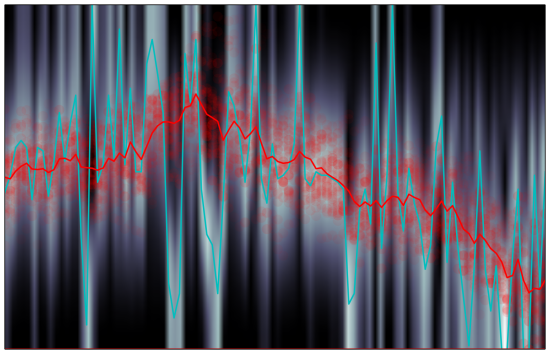
By looking at the different cases in Figure 5.7, we can see how the different particles react to an outlier observation. For the time frame plotted, the observation is located at the base of the hemisphere, while the current state estimate is at the top of the hemisphere. Note the different particle weights for the three cases (as indicated by the size of the particles). For decreasing values of $\Pr(c_t = 1)$, the weights become more uniform, and the estimates are therefore less affected. This reaffirms the analysis given in Section 5.3.5

Thus we see that by using a switching state space model with a "spike and slab" emission distribution we are able to deal with inaccurate observations that will result in real world reverberant environments. This improves performance over the standard vMFPF as can be seen by comparing Figure 5.2 with Figure 5.7 or by comparing Figure 5.7a with Figure 5.7c.



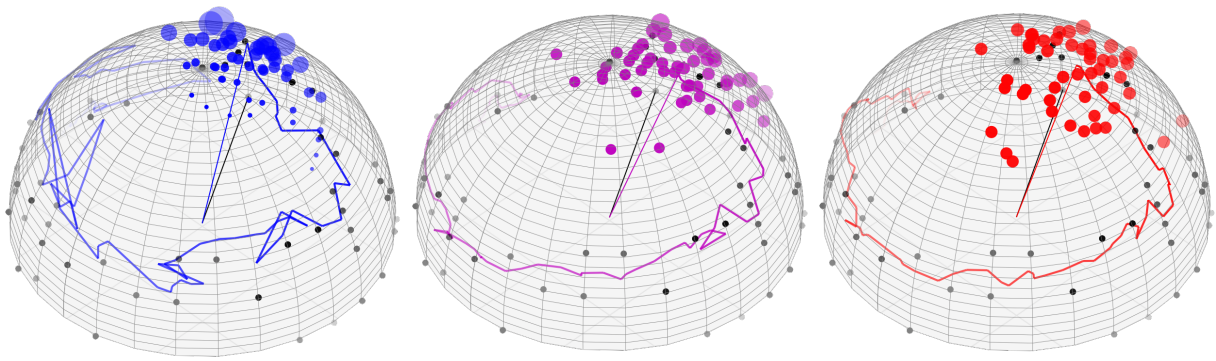
(a) $\Pr(c_t = 1) = 1$

(b) $\Pr(c_t = 1) = .7$



(c) $\Pr(c_t = 1) = .4$

Figure 5.6: von Mises switching particle filter tracking. We see the performance of the switching particle filter for various values of the "spike" prior $\Pr(c_t = 1)$ when tracking on one half of the unit circle. The format of the plot follows that given in Figure 5.5. When $\Pr(c_t = 1) = 1$, as shown in (a), the algorithm reduces to the standard particle filter algorithm. 30 particles were used, with $\kappa_u = 100$ and $\kappa_v = 5$. Data was recorded on a Playstation Eye.



(a) $\Pr(c_t = 1) = 1$ (b) $\Pr(c_t = 1) = .7$ (c) $\Pr(c_t = 1) = .4$

Figure 5.7: von Mises-Fisher switching particle filter tracking. We see the performance of the switching particle filter for various values of the "spike" prior $\Pr(c_t = 1)$ when tracking on the unit sphere. DOA observations are represented by black dots, while the state estimate path is given along with particles, whose sizes indicate their weight. When $\Pr(c_t = 1) = 1$, as shown in (a), the algorithm reduces to the standard particle filter algorithm. 50 particles were used, with $\kappa_u = 100$ and $\kappa_v = 5$. Data was recorded on a Dev Audio Microcone.

5.4 von Mises-Fisher Steered Response Power Particle Filter (SRPPF)

While the vMFSPF gives an improvement over the standard vMFPF, it still has one major shortcoming. By using point estimates for the DOA observations, it discards a wealth of information inherent in the signal and then compensates for it by a more elaborate system model. We have seen that if the system model is setup to represent the data, it can give decent performance. However, this suggests that using a more faithful distribution for our observation emissions could be helpful.

5.4.1 Model Definition

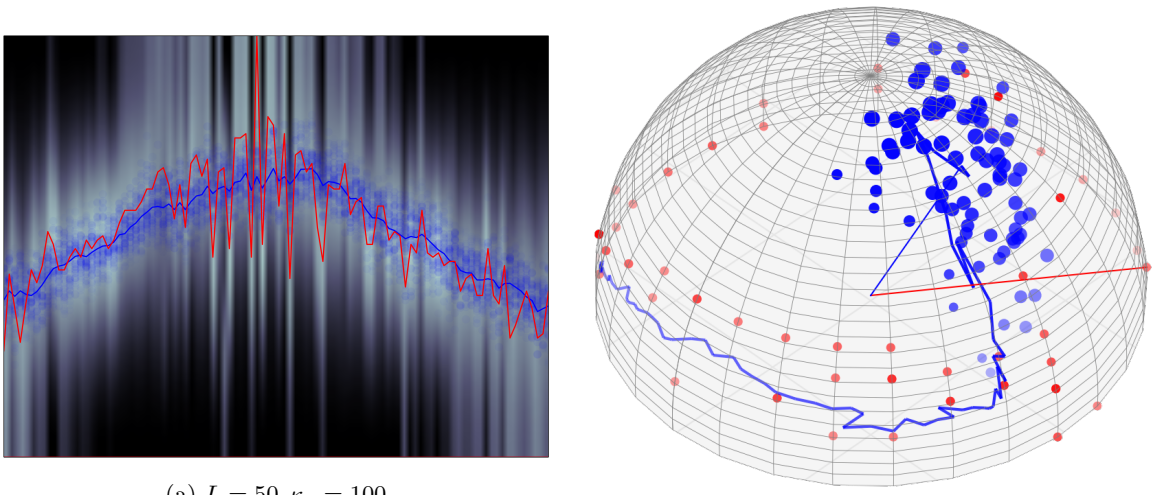
In the vMPF model we defined our emission distribution as

$$p(\mathbf{z}_t|\mathbf{x}_t) = v\mathcal{MF}(\mathbf{z}_t; \mathbf{x}_t, \kappa_v), \tag{5.45}$$

where we took $\mathbf{z}_t$ to be a single DOA observation. However, we can instead take $\mathbf{z}_t$ to be the signals recorded at the microphones for time frame t. Now we simply need a way of evaluating the likelihood of the signals given a certain DOA $\mathbf{x}_t$. Such methods were discussed in Section 3.1.3 and Section 3.1.4, and so we use some definition of $\mathcal{L}(\cdot)$ as discussed:

$$p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{L}(\mathbf{x}_t; \theta_v). \tag{5.46}$$

where θ_v is some set of parameters that can be used to tune the likelihood function $\mathcal{L}(\cdot)$. For example, this could contain a shaping coefficient value k for the shaping function $\Phi(\cdot)$ as discussed in Section 3.1.3. This



(a) $L = 50, \kappa_u = 100$

(b) $L = 80, \kappa_u = 100$

Figure 5.8: von Mises-Fisher SRP particle filter tracking. $\mathcal{L}(\cdot) = \mathcal{L}_{\text{DS}}(\cdot)$. For (a), conditions were the same as in Figure 5.1. For (b), conditions were the same as in Figure 5.2. In (b), red dots indicate the peaks in $\mathcal{L}(\mathbf{x}_t)$, which are what would have been used as point observations previously. However, they are only shown here for reference.

gives us our new model:

$$\mathbf{x}_t \sim v\mathcal{MF}(\mathbf{x}_{t-1}, \kappa_u), \tag{5.47}$$

$$\mathbf{z}_t \sim \mathcal{L}(\mathbf{x}_t; \theta_v). \tag{5.48}$$

There are a few immediate benefits of this model. The most obvious is that we have replaced our point observations with our true signal observations. In this way, we have included all available information in our model and are using the raw data to calculate our likelihoods instead of using a single output of another calculation relying on that data.

Secondly, we no longer need to discretize our DOA space. Because our particles lie in a continuous state space and we can evaluate $\mathcal{L}(\mathbf{x})$ for any $\mathbf{x}$, there is no reason to sample the space. This follows from the fact that we no longer need to calculate a DOA to use as our observation. This can help avoid errors brought on by the discretization and reduce computational load, as described in [20].

5.4.2 Performance

Examples of the tracking performance of the SRPPF model are given in Figure 5.8. We see that in both the two-dimensional case and three-dimensional case results are improved from the vMPF, with the three-dimensional performance in Figure 5.8b being an immense improvement over the results displayed in Figure 5.2. Comparing these figures also provides a good illustration of why the performance has improved.

As mentioned before, Figure 5.2 shows what can happen under the von Mises-Fisher emission model when

the DOA point observation is far from the current state estimate. However, in Figure 5.8b, we see that the current DOA point observation is also far from the state estimate, yet the particle weights are far more uniform. This is because we are using the actual DOA likelihood function $\mathcal{L}(\mathbf{x}_t)$ to evaluate likelihoods, which may have more than one peak. Thus, even if we have a large likelihood on the other side of the hemisphere, if the likelihood is also large near the current state estimate, the particles will not erroneously jump across the space. This helps to smooth our estimates, as can be seen.

CHAPTER 6

VIDEO RECORDING

To use the methods outlined in previous chapters, we must convey the information gathered from our microphone array to the camera so the speaker may be recorded properly. The inherent difficulty in this process is that we will be provided some estimate of direction from the microphone array, but will have no sense of distance from the microphone array to the source. Therefore we must constrain our problem in some way to locate the speaker in the space of the room.

6.1 Constrained Lecture Space

As discussed in Section 4.1, it is often possible to constrain the speaker to lie on a plane at the front of the room. If this is the case, we can use Equation (4.3) to estimate the speaker's location $\mathbf{s}(\mathbf{v})$ in the room's coordinates given a DOA estimate $\mathbf{v}$ from the microphone array. Using this and the knowledge of the camera's location, we can correctly turn the camera to point to this position in the room. For a demonstration of this method see Figure 6.1.

While this approach may work pretty well when the lecturer is the only one who speaks, we will run into problems if someone in the audience contributes to the discussion. There are two possible problems. In the first case, if the direction to the audience member is on the same side of the room as the speaker from the perspective of the microphone array, then the DOA of the audience member can be mapped to a point on the speech plane. Unfortunately the point will likely be very incorrect, as the audience member will not in fact lie on the plane. In the second case, the audience member will be in a direction from the point of view of the microphone array such that the ray extending from the microphone in that direction never intersects the speaker plane, and our previous method will not be able to map it to the plane at all.

One way to remedy this is by defining multiple planes on which speakers may lie. For example, in addition to the lecturer plane in the front of the room, we may define a plane parallel with the classroom floor that is roughly at the height of the average speaker's head. Now when mapping a DOA to a point in the room, we can attempt to do so for both planes, and take the point that is closest. This method can be repeated until all feasible locations for sound sources are covered by some plane.

While this method may address some of our problems, is has a few drawbacks. First of all, the performance of the system will depend on the fidelity of our model and the tendencies of the speakers in the room to

comply with their expected behaviors. Additionally, it requires that we set up different constraints for each unique room the system is used in. While this would be a one-time task, it could become quite tedious if the system were implemented on a large scale.

6.2 Coincident Recorders

Another approach is to avoid constraints on the speaker's locations and instead enforce some constraint on the positions of the camera and microphone array. The obvious way to do this is to place the microphone array and camera as close to each other as possible. By doing this, we can simply point the camera in the same direction as our DOA and if the array and camera are close enough and the speaker is far enough away, the error should be relatively small.

The problem with this is it greatly reduces flexibility in refining the system, as the camera and microphones are now coupled. If we were to find that certain locations in the room were great for array processing but not for filming, or vice versa, we would be out of luck.

6.3 More Arrays

A final approach would be to avoid either of the aforementioned constraints and instead double the number of estimates. If we were to make use of a second array to provide an additional DOA estimate, it would be possible to triangulate the source position within some error. Unfortunately, this is not always practical as microphone arrays are not necessarily cheap enough to double their use in a system.

Figure 6.1: Camera demonstration. Demonstration of system as described. A PTZ Camera in Siebel Center at the University of Illinois at Urbana-Champagin was used to film a speaker. The camera is located on the ceiling of the room, and the microphone array on a desk two meters from the front of the room. A Dev Audio Microcone array was used with the vMFSPF method for tracking and the constrained lecture space method for locating the source in the room.

CHAPTER 7

CONCLUSION

7.1 Overview

We have described several microphone array processing methods for tracking a speaker in a lecture setting. We first examined effective methods of computing direction of arrival likelihoods for a given set of signals. We then used these techniques to apply several different recursive Bayesian filtering approaches to our lecturer tracking problem.

By constraining the lecturer to a plane in the auditorium or classroom, we were able to make use of certain established techniques including grid-based recursive estimation methods and Kalman filtering. While grid-based methods worked only for a narrow range of motion, Kalman filtering performed well across most of the search space provided that the lecturer remained near the constraint plane. Because of their simplicity, these approaches could prove useful in cases where the stipulations on the lecturer's location are satisfied and the lecturer's range of motion is limited.

Additionally, we investigated unconstrained tracking methods where we performed direction of arrival estimation directly on the unit sphere. To do this we made use of sequential Monte Carlo methods, or Particle filtering. We found that using a von Mises-Fisher particle filter with von Mises-Fisher transition and emission distributions resulted in problems with noise and inaccurate observations in real world environments. To address this, we derived the von Mises-Fisher switching particle filter that used a switching state space model to model the emission distribution. This was able to capture the possibility of inaccurate observations in the system model and thus performed well in a real reverberant environment. We also investigated a von Mises-Fisher steered response power particle filter that utilized the true steered response power to evaluate likelihoods and that didn't require discretization of our search space. This model was also able to perform well in a reverberant environment.

Finally we proposed different methods of utilizing localization information from a microphone array with a camera to properly film a lecturer. Again we found that by assuming certain constraints on the lecturer's movement we were able to arrive at an effective solution. We then combined the discussed techniques to track a speaker as they moved about the front of a classroom in real-time.

7.2 Future Work

All of the focus in this project went into microphone array methods for tracking. However, since both a camera and microphone array are available, it could be advantageous to combine video and audio tracking methods to create a more robust procedure. Additionally, it would be important in a real system to address problems of multiple simultaneous sound sources. We focused on tracking single sources and attributed all other sound to noise. However, in a discussion setting there would be an exchange between multiple sources and the tracking methods would need to account for this. A difficulty here would be that the number of relevant sources would be changing throughout time, yet there may again be constraints provided by the lecture setting that would simplify the scenario and help assess when such an approach would be necessary. There is much room for improvement in the automation of the lecture monitoring systems currently being developed, and such improvements could be of great help.

REFERENCES

[1] L. A. Rowe, D. Harley, P. Pletcher, and S. Lawrence, "Bibs: A lecture webcasting system," Center for Studies in Higher Education, UC Berkeley, University of California at Berkeley, Center for Studies in Higher Education, 2001. [Online]. Available: http://EconPapers.repec.org/RePEc:cdl:cshedu:qt48q7t01w

[2] S. Mukhopadhyay and B. Smith, "Passive capture and structuring of lectures," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. ACM, 1999, pp. 477–487.

[3] Q. Liu, Y. Rui, A. Gupta, and J. J. Cadiz, "Automating camera management for lecture room environments," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '01. New York, NY, USA: ACM, 2001. [Online]. Available: http://doi.acm.org/10.1145/365024.365310 pp. 442–449.

[4] M. Bianchi, "Automatic video production of lectures using an intelligent and aware environment," in *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '04. New York, NY, USA: ACM, 2004. [Online]. Available: http://doi.acm.org/10.1145/1052380.1052397 pp. 117–123.

[5] C. Zhang, Y. Rui, J. Crawford, and L. wei He, "An automated end-to-end lecture capturing and broadcasting system," in *In Proceedings of the 13th Annual ACM international Conference on Multimedia - MULTIMEDIA'05*. ACM Press, 2005, pp. 808–809.

[6] H.-P. Chou, J.-M. Wang, C.-S. Fuh, S.-C. Lin, and S.-W. Chen, "Automated lecture recording system," in *System Science and Engineering (ICSSE), 2010 International Conference on*. IEEE, 2010, pp. 167–172.

[7] I. J. Tashev, *Sound Capture and Processing: Practical Approaches*. John Wiley & Sons, 2009.

[8] D. E. Dudgeon, "Fundamentals of digital array processing," *Proceedings of the IEEE*, vol. 65, no. 6, pp. 898–904, 1977.

[9] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, ser. Springer Topics in Signal Processing. Springer, 2010.

[10] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[11] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Signal Processing, IEEE Transactions on*.

[12] N. Bergman, "Recursive Bayesian estimation," *Linköping Studies in Science and Technology, Dissertation*, no. 579, pp. 21–43, 1999.

[13] C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320–327, 1976.

[14] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 375–378.

[15] M. Omologo and P. Svaizer, "Use of the cross-power-spectrum phase in acoustic event location," *IEEE Speech and Audio Processing, IEEE Transactions on.*

[16] R. Duraiswami, D. Zotkin, and L. S. Davis, "Active speech source localization by a dual coarse-to-fine search," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, vol. 5. IEEE, 2001, pp. 3309–3312.

[17] J. Dmochowski, J. Benesty, and S. Affes, "Fast steered response power source localization using inverse mapping of relative delays," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on.* IEEE, 2008, pp. 289–292.

[18] J. H. DiBiase, "A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays," Ph.D. dissertation, Brown University, 2000.

[19] M. S. Brandstein, J. E. Adcock, and H. F. Silverman, "A closed-form location estimator for use with room environment microphone arrays," *IEEE Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 1, pp. 45–50, 1997.

[20] D. B. Ward and R. C. Williamson, "Particle filter beamforming for acoustic source localization in a reverberant environment," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, vol. 2. IEEE, 2002, pp. II–1777.

[21] S. T. Birchfield and D. K. Gillmor, "Fast Bayesian acoustic localization," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, 2002.

[22] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[23] G. Bishop and G. Welch, "An introduction to the Kalman filter," 2001. [Online]. Available: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[24] C. M. Bishop et al., *Pattern Recognition and Machine Learning.* Springer New York, 2006, vol. 1.

[25] U. Klee, T. Gehrig, and J. McDonough, "Kalman filters for time delay of arrival-based source localization," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, 2006.

[26] J. Traa and P. Smaragdis, "A wrapped Kalman filter for azimuthal speaker tracking," *Signal Processing Letters, IEEE*, vol. 20, no. 12, pp. 1257–1260, Dec 2013.

[27] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[28] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American statistical association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[29] S. P. Brooks and S. P. Brooks, "Markov chain Monte Carlo method and its application. statistician 47, 69–100. regression in capture–recapture modeling 697," *Statistical Science*, pp. 357–376, 1998.

[30] W. K. Hastings, "Monte Carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[31] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, and G. Ridgeway, "Clustering on the unit hypersphere using von mises-fisher distributions." *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.

[32] J. Traa, "Multichannel source separation and tracking with phase differences by random sample consensus," M.S. thesis, University of Illinois at Urbana-Champaign, 2013.

[33] J. Glover and L. P. Kaelbling, "Tracking 3-d rotations with the quaternion bingham filter," MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Tech. Rep., 2013. [Online]. Available: http://dspace.mit.edu/bitstream/handle/1721.1/78248/MIT-CSAIL-TR-2013-005.pdf?sequence=1

[34] F. Caron, M. Davy, E. Duflos, and P. Vanheeghe, *IEEE Signal Processing, IEEE Transactions on.*

[35] K. P. Murphy, "Dynamic Bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, 2002.

[36] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.

[37] K. Kawamoto, "Grid-based rao-blackwellisation of particle filtering for switching observation models," in *Information Fusion (FUSION), 2012 15th International Conference on*, July 2012, pp. 143–148.